UNIVERSITE DE MONTREAL

METHODE DE PANNEAUX POUR ECOULEMENTS TRANSSONIQUES

par

Christian MASSON

DEPARTEMENT DE GENIE MECANIQUE

ECOLE POLYTECHNIQUE

MEMOIRE PRESENTE EN VUE DE L'OBTENTION

DU GRADE DE MAITRE ES SCIENCES APPLIQUEES (M.Sc.A.)

Juillet 1989

c Christian Masson 1989

ıal Library
nada

Bibliothèque nationale
du Canada

ıdian Theses Service

Service des thèses canadiennes

, Canada
N4

e author has granted an irrevocable non-
:lusive licence allowing the National Library
Canada to reproduce, loan, distribute or sell
ɔies of his/her thesis by any means and in
' form or format, making this thesis available
nterested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

e author retains ownership of the copyright
his/her thesis. Neither the thesis nor
ɔstantial extracts from it may be printed or
erwise reproduced without his/her per-
ssion.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-52704-8

anadä

UNIVERSITE DE MONTREAL

ECOLE POLYTECHNIQUE

Ce mémoire intitulé :

METHODE DE PANNEAUX POUR ECOULEMENTS TRANSSONIQUES

présenté par : Christian Masson

en vue de l'obtention du grade : M.Sc.A.

a été dûment accepté par le jury d'examen constitué de :

M.Ricardo Camarero, Ph.D.,Président.

M.Ion Paraschivoiu, Ph.D.,Directeur de recherche

M.Zhigang Fang, Ph.D.

à Marie...

SOMMAIRE

Ce mémoire porte sur l'application d'une méthode numérique et le développement d'un logiciel de calcul pour les écoulements transsoniques autour de corps bidimensionnels. Une comparaison avec des résultats expérimentaux et un programme commercial éprouvé confirme la validité de nos résultats.

L'approche utilisée est fondée sur la méthode des panneaux transsonique qui permet de combiner la polyvalence des méthodes de panneaux classiques aux techniques de solution appropriées au régime transsonique non linéaire.

Cette nouvelle approche est la seule, actuellement, qui permette de traiter facilement des configurations très complexes. En effet, contrairement aux méthodes traditionnelles (éléments finis, volumes finis et différences finies), elle ne nécessite pas la génération d'un maillage adapté. Un maillage rectangulaire peut être utilisé quel que soit la géométrie à étudier puisque aucune condition aux frontières n'est appliquée sur les points du maillage.

Dans ce travail, nous présentons une methode 2-D qui représente une première étape dans le développement d'un algorithme 3-D et d'un programme axé sur l'analyse et le design d'ailes d'avion. Cette formulation intégrale

permettra d'évaluer les caractéristiques aérodynamiques d'un corps quelconque jusqu'au régime transsonique non linéaire où il existe des solutions discontinues (Chocs).

Le grand avantage de cette approche est de fournir aux entreprises canadiennes de l'aéronautique un outil pratique d'analyse de géométries complexes utilisable à court terme et ainsi répondre à un besoin pressant de cette industrie.

ABSTRACT

The present thesis concerns the application of a numerical method and the development of a computer code for the design and analysis of wing sections in transonic regime. A comparison with experimental data and commercial computer code validates our results.

A new approach, called the Field Panel Method, is used. It permits to combine the advantages of panel methods with solution techniques appropriate for the non linear transonic regime.

The methods currently used for transonic flows are field methods (finite-elements, finite volume and finite-difference methods) and they need to construct a grid fitted to an infinite computational domain and its boundary which could be a major, if not impossible, task for complex multi-body configurations such as wing mounted nacelles.

This new approach, with its rectangular grid, is actually the only one which offers the possibility of treating very complex configurations since no boundary condition is applied on the grid points.

In this work, we present a 2-D method which is a first step in the development of a 3-D numerical method and computer code for the design and analysis of wings. The chosen integral

formulation will permit to evaluate aerodynamic characteristics of arbitrary configurations in flows up to the non linear transonic regime where discontinuous solutions (shock waves) can occur.

The main advantage of this approach is to supply rapidly to the aeronautic canadian manufacturers a design tool for complex configurations.

REMERCIEMENTS

L'ensemble des travaux dont les résultats font l'objet du présent mémoire a été réalisé au Département de Génie Mécanique, section Aérothermique avec l'équipe du Professeur Ion Paraschivoiu. Cette étude a été réalisée en partie grâce à une bourse du CRSNG.

Je tiens en tout premier lieu à exprimer ma reconnaissance à Monsieur Fassi Kafyeke, ingénieur chez Bombardier/Canadair Inc., qui m'a proposé le sujet de cette recherche.

Je remercie le Professeur Ion Paraschivoiu qui a su bien me diriger tout au long de ce travail.

Je ne peux passer sous le silence les judicieux conseils de Messieurs Louis Lamarche et Olivier Marchand.

Merci enfin à Mesdames Sylvie Peltier et Brigitte Hayeur qui ont su rendre ma tâche plus facile...

TABLE DES MATIERES

SOMMAIRE v

ABSTRACT vii

REMERCIEMENTS ix

LISTE DES FIGURES xii

NOMENCLATURE xiii

CHAPITRE 1 : INTRODUCTION 1

CHAPITRE 2 : FORMULATION 4

CHAPITRE 3 : METHODE NUMERIQUE 7

3.1- Discrétisation 7

3.1.1- Intégrales de surface 9

3.1.2- Intégrale de volume 11

3.1.3- Méthode de panneaux classique 12

3.1.4- Méthode de panneaux transsonique 15

3.1.5- Coefficient de pression 16

3.1.6- Coefficient de portance 17

3.1.7- Distribution des panneaux 17

3.1.8- Détermination des points intérieurs 19

3.2- Algorithme 21

3.2.1- Procédé itératif 21

3.2.2- Méthode de panneaux avec doublets linéaires . 22

3.2.3- Schéma tourné 26

3.3- Description du programme 31

3.3.1- Données 31

3.3.2- Résultats 32

3.3.3- Sous-routines 33

CHAPITRE 4 : DISCUSSION DES RESULTATS 36

CHAPITRE 5 : CONCLUSION 42

REFERENCES 45

ANNEXE 1 : EQUATION DU POTENTIEL COMPLET 49

ANNEXE 2 : INTEGRALES DE SURFACE 52

ANNEXE 3 : INTEGRALE DE VOLUME 54

ANNEXE 4 : EQUIVALENCE DOUBLET-TOURBILLON 61

ANNEXE 5 : PROGRAMME 62

FIGURES 92

LISTE DES FIGURES

Fig. 1 : Définition du problème.

Fig. 2 : Solution intégrale.

Fig. 3 : Discrétisation du corps.

Fig. 4 : Panneau de doublet.

Fig. 5 : Domaine de calcul.

Fig. 6 : Pavé de source.

Fig. 7 : Bord de fuite

Fig. 8 : Parcours équivalent.

Fig. 9 : Coefficient de pression, $M_\infty = 0.703$, $\alpha = 0.000$.

Fig. 10 : Nombre de Mach local, $M_\infty = 0.703$, $\alpha = 0.000$.

Fig. 11 : Coefficient de pression, $M_\infty = 0.803$, $\alpha = 0.000$.

Fig. 12 : Nombre de Mach local, $M_\infty = 0.803$, $\alpha = 0.000$.

Fig. 13 : Coefficient de pression, $M_\infty = 0.754$, $\alpha = 0.990$.

Fig. 14 : Nombre de Mach local, $M_\infty = 0.754$, $\alpha = 0.990$.

Fig. 15 : Coefficient de pression - Euler, $M_\infty = 0.703$, $\alpha = 0.000$.

Fig. 16 : Coefficient de pression - Euler, $M_\infty = 0.803$, $\alpha = 0.000$.

Fig. 17 : Coefficient de pression - Quasi-conservatif, $M_\infty = 0.803$, $\alpha = 0.000$.

Fig. 18 : Pavé de source, point à l'intérieur du pavé.

NOMENCLATURE

a	vitesse du son, m/s, éqn. (75)
C_L	coefficient de portance, éqn. (38)
C_p	coefficient de pression, éqn. (86)
c	corde du profil, m
d	longueur, m, voir Fig. 6
G	fonction de Green, éqn. (9)
L	portance, N
l	longueur d'un panneau, m
M	nombre de Mach, éqn. (6)
N	nombre de panneaux
Nx	nombre de pavés en x
Ny	nombre de pavés en y
R	position du point P, m, voir Fig. 2
R_1, R_2	longueur, m, voir Fig. 6
R_3, R_4	
r	distance, m, voir Fig. 2
r_s	position du point d'intégration, m, voir Fig. 2
S	surface, m^2
u	composante de la vitesse dans la direction de x
v	composante de la vitesse dans la direction de y
x	coordonnée orthogonale, m
x_{choc}	position du choc, m
y	coordonnée orthogonale, m
Δx	longueur d'un pavé en x, m

Δy longueur d'un pavé en y, m

$\vec{s}$ vecteur dans la direction de V

$\vec{V}$ vecteur vitesse

$\vec{n}$ vecteur normal

α angle d'attaque, rad.

β angle, rad., voir Fig. 4

ξ variable d'intégration, éqn. (25)

Φ potentiel

Γ circulation, m^2/s, éqn. (21)

γ rapport des chaleurs spécifiques

Ω volume, m_3

θ_1, θ_2 angle, rad., voir Fig. 6

θ_3, θ_4

INDICES INFERIEURS

∞ Relatif à l'écoulement non perturbé

i Relatif au $i^{ième}$ panneau

j Relatif au $j^{ième}$ panneau

n Relatif au $n^{ième}$ pavé dans la direction x

m Relatif au $m^{ième}$ pavé dans la direction y

t Relatif à la tangente

INDICES SUPERIEURS

p Induit par les panneaux de doublet

f Induit par les sources du champs

* Relatif au panneau ou au pavé

CHAPITRE 1

INTRODUCTION

Le sujet de ce mémoire est situé dans le domaine de l'aérodynamique numérique et a pour objectif d'appliquer une méthode numérique et un logiciel de calcul pour les écoulements transsoniques autour de corps bidimensionnels.

A l'heure actuelle, la méthode la plus utilisée en aérodynamique numérique est la méthode des panneaux ou méthode intégrale. En régime permanent, les écoulements non-visqueux incompressibles et irrotationnels sont gouvernés par l'équation de Laplace qui est linéaire. En appliquant le théorème de Green, on peut démontrer que le potentiel (et donc le champ de vitesse) dans tout l'espace autour d'un corps peut être représenté par une répartition de singularités distribuées sur la surface du corps uniquement. D'abord limitée au cas d'écoulements subsoniques, la méthode des panneaux a été étendue au cas supersonique par la résolution de l'équation de Prandlt-Glauert. Actuellement tous les manufacturiers ont développé leur propre logiciel de calcul basé sur une méthode intégrale. Une bonne étude comparative des différents modèles existants a été effectuée par Sytsma et al. [1]

Les méthodes intégrales décrites précédemment ne s'appliquant qu'à des écoulements linéaires subsoniques ou supersoniques, une approche très différente est utilisée pour les écoulements transsoniques non linéaires. Elle est basée sur des méthodes de champs (différence-finie, volume fini ou élément fini) appliquées aux équations des petites perturbations, du potentiel complet, d'Euler ou de Navier-Stokes selon le degré de sophistication du modèle ou le type d'application. Une revue des différents modèles existants a été réalisée par Jameson [2]. La grande différence entre les modèles numériques se situe au niveau de l'algorithme de capture de chocs. Un des premiers algorithmes de capture de chocs fut réalisé par Murman [3] et est basé sur l'équation des petites perturbations. Le principe consistait à utiliser une représentation centrée de certaines dérivées dans les régions subsoniques et une représentation amont pour les points supersoniques. Jameson [4] dériva un algorithme semblable mais basé sur l'équation du potentiel complet. C'est le schéma tourné. Ces deux schémas violaient légèrement la conservation de la masse [2] et il a été démontré que l'ajout de viscosité artificielle explicite sous une forme conservative [2] respecte la conservation de la masse.

Tous ces modèles (sauf ceux appliqués à l'équation des petites perturbationsnécessitent la génération d'un

maillage adapté à la configuration étudiée ce qui constitue une tâche très difficile pour des géométries très complexes (par exemple aile-nacelle et/ou aile-fuselage). Cela est pénalisant pour l'application industrielle des logiciels de calcul existants notamment lors de la phase de conception d'un appareil. C'est pourquoi une nouvelle méthode a été récemment développée [5-8]. Elle consiste à étendre la méthode des panneaux classique au cas transsonique. Ceci permet de tirer profit des avantages de la méthode intégrale (traitement de géométries complexes, de conditions de design, etc.) pour les écoulements transsoniques. Cette approche très attrayante est connue sous le nom de méthode de panneaux transsonique. Johnson et al. [5] de Boeing et Sinclair [6] de British Aerospace ont déjà mis au point des codes 2-D et travaillent actuellement sur le cas 3-D [7,8].

CHAPITRE 2

FORMULATION

Pour les nombres de Reynolds élevés (de l'ordre de 10^7 et plus), l'écoulement autour d'un corps aérodynamique peut être considéré non-visqueux et irrotationnel. L'équation gouvernant un tel écoulement (écoulement potentiel) est l'équation du potentiel complet [9] (voir annexe 1 pour la dérivation de cette équation) :

$$\nabla^2 \Phi = \frac{1}{2a^2} \nabla\Phi \cdot \nabla |\nabla\Phi|^2 \qquad (1)$$

considérant les conditions aux limites suivantes: (Fig. 1)

$$\vec{V} \longrightarrow \vec{V}_\infty \qquad \text{sur } S_\infty \text{ (à l'infini)} \qquad (2)$$

$$\vec{n} \cdot \nabla\Phi = 0 \qquad \text{sur } S_c \text{ (sur le corps)} \qquad (3)$$

Du fait que l'écoulement est irrotationnel, la vitesse $\vec{V}$ dérive d'un potentiel Φ avec:

$$\vec{V} = \nabla\Phi \qquad (4)$$

Si $\vec{s}$ désigne la direction de la vitesse $\vec{V}$, l'équation mixte elliptique-hyperbolique (1) peut être réécrite sous la forme suivante:

$$\nabla^2 \Phi = M^2 \ \Phi_{ss} \qquad (5)$$

où Φ_{ss} est la seconde dérivé de Φ dans le sens de la vitesse $\vec{V}$.

$$M^2 = \frac{|\vec{V}|^2}{a^2} \qquad (6)$$

$$a^2 = \frac{1}{M_\infty^2} + \frac{\gamma-1}{2}\,(1-|\nabla\Phi|^2) \qquad (7)$$

(voir annexe 1 pour la dérivation de l'équation (7))

En utilisant les fonctions de Green, la solution intégrale de l'éqn. (5) s'écrit [10]:

$$\Phi = \Phi_\infty - \int_{S_c+S_s} \Phi\ \vec{n}\cdot\nabla G(r)\ ds + \int_\Omega M^2\ \Phi_{ss}\ G(r)\ d\Omega \qquad (8)$$

où $G(r) = \dfrac{1}{2\pi}\ln(r)$ (9)

et $r = |\vec{r}_s - \vec{R}|$ d'après la Fig. 2) (10)

L'équation (8) signifie que la solution Φ peut être représentée par une distribution de doublets sur la surface du corps et du sillage (S_c+S_s) et de sources sur le volume entourant le corps (Ω)

Pour un écoulement incompressible, le terme $M^2\ \Phi_{ss}$ est nul, alors l'équation (8) sans l'intégrale de volume est résolue directement en utilisant une méthode de panneaux classique [11,12,13]. Pour les écoulements compressibles avec ou sans choc, le terme $M^2\ \Phi_{ss}$ n'est pas nul et la solution est déterminée itérativement en solutionnant l'équation (8) complète. L'évaluation de l'intégrale de volume permet de calculer les effets compressibles tandis que l'application de l'équation (8) sur S_c assure le respect des conditions aux frontières sur les parois du corps. Le

volume Ω est un domaine infini. En pratique, comme $M^2\ \Phi_{ss}$ représente les effets compressibles, ce terme tend vers zéro rapidement lorsque l'on s'éloigne du corps [14]. Il est donc suffisant de ne considérer que la région près du corps lors de l'évaluation de l'intégrale de volume.

Pour faciliter les manipulations subséquentes, nous allons définir deux potentiels Φ^p et Φ^f tel que:

$$\Phi = \Phi^p + \Phi^f \tag{11}$$

Φ_p est le potentiel induit par l'écoulement uniforme et les doublets sur le profil tandis que Φ_f est le potentiel induit par les sources de la région Ω.

où $$\Phi^p = \Phi_\infty - \int_{S_c+S_s} \Phi\ \vec{n}\cdot\nabla G(r)\ ds \tag{12}$$

$$\Phi^f = \int_\Omega M^2\ \Phi_{ss}\ G(r)\ d\Omega \tag{13}$$

La formulation décrite ci-desssus nous permet d'obtenir le potentiel de vitesse Φ. A partir de cette solution, on obtient facilement la distribution de pression sur le corps en utilisant la relation suivante (voir annexe 1 pour la dérivation de cette équation):

$$C_p = \frac{2}{\gamma M_\infty^2}\left[\left[\frac{\gamma-1}{2}M_\infty^2(1-|\nabla\Phi|^2)+1\right]^{\frac{\gamma}{\gamma-1}}-1\right] \tag{14}$$

CHAPITRE 3

METHODE NUMERIQUE

3.1 - DISCRETISATION

La solution discrète du problème considéré est obtenue en utilisant une méthode de panneaux. Cette dernière a été choisie pour sa très grande facilité à résoudre l'équation de Laplace autour d'un corps complexe. Elle est basée sur la solution de l'équation (8). Cette équation a été obtenue par la technique des fonctions de Green.

Nous allons tout d'abord faire le développement qui nous permet de passer de la formulation du problème (éqns (1)-(3)) à l'équation (8).

En utilisant la méthode des fonctions de Green, la solution du problème peut s'écrire sous la forme intégrale suivante :

$$\Phi(P) = \int_{S_T} [(\vec{n}\cdot\nabla\Phi)\, G(r) - \Phi\, \vec{n}\cdot\nabla G(r)]\, ds + \Phi^f(P) \qquad (15)$$

où $G(r)$ est donnée par la relation (9)

$\Phi^f(P)$ est donné par la relation (13)

$$S_T = S_c + S_\infty + 2S_s \qquad (16)$$

Or, en appliquant la condition (2) pour un point P sur S_∞, on trouve que:

$$\Phi(P) \approx \Phi_\infty(P) = x(P)\, \vec{V}_\infty\cdot\vec{i} + y(P)\, \vec{V}_\infty\cdot\vec{j} \qquad (17)$$

Alors l'équation (15) peut être réécrite sous la forme suivante:

$$\Phi(P) = \Phi_\infty(P) + \Phi^f(P) + \int_{S_c+2S_s} [(\vec{n}\cdot\nabla\Phi)G(r) - \Phi\,\vec{n}\cdot\nabla G(r)]ds \tag{18}$$

Comme nous étudions un milieu continu et que la normale sur la branche inférieure de S_s est égale mais de signe opposé à la normale sur la branche supérieure de S_s, on peut dire que:

$$\vec{n}\cdot\nabla\Phi\,|_{bas} = -\vec{n}\cdot\nabla\Phi\,|_{haut} \quad \text{sur } S_s \tag{19}$$

En utilisant cette dernière équation et la condition (3), la relation (18) devient:

$$\Phi(P) = \Phi_\infty(P) + \Phi^f(P) - \int_{S_c+2S_s} \Phi\,\vec{n}\cdot\nabla G(r)\,ds \tag{20}$$

Cette dernière relation est identique à la relation (8).

Par définition, la circulation Γ autour du corps est:

$$\Gamma = \int_C \vec{V}\cdot d\vec{l} \tag{21}$$

où C est un parcours fermé autour du corps. Egalement par définition, la différence dans Φ entre deux points P_1 et P_2 est

$$\Phi(P_2) - \Phi(P_1) = \int_{P_1}^{P_2} \nabla\Phi\cdot d\vec{l} = \int_{P_1}^{P_2} \vec{V}\cdot d\vec{l} \tag{22}$$

Si P_1 est situé sur S_s en bas et P_2 sur S_s en haut, alors on

peut écrire que:

$$\Phi(P_2) - \Phi(P_1) = \int_{P_1}^{P_2} \vec{V} \cdot d\vec{l} = \Gamma \tag{23}$$

La dernière relation s'applique peut importe la position des points P_1 et P_2 sur S_c. On en conclut donc que la discontinuité en Φ est constante et égale à Γ sur S_s. Utilisant cette affirmation, l'équation intégrale (20) peut être réécrite sous la forme suivante:

$$\Phi(P) = \Phi_\infty(P) + \Phi^f(P) - \int_{S_c} \Phi \, \vec{n} \cdot \nabla G(r) \, ds - \Gamma \int_{S_s} \vec{n} \cdot \nabla G(r) \, ds \tag{24}$$

3.1.1 - INTEGRALES DE SURFACE

La seconde étape est la discrétisation des intégrales de surface de l'équation (24). Tout d'abord, on remplace la surface courbe du corps par un polygone à N cotés: on choisit un ensemble de points(noeuds) sur le corps que l'on réunit pour former les panneaux. On obtient ainsi N panneaux et N+1 noeuds: les points 1 et N+1 sont au même endroit pour un corps fermé. (Fig. 3). Les noeuds sont numérotés dans le sens horaire en partant du bord de fuite et le $j^{ième}$ panneau est situé entre les noeuds j et j+1. La surface S_s part du bord de fuite (noeud 1) et continue jusqu'à l'infini en longeant l'axe des x. C'est le $N+1^{ième}$ panneau. Ensuite, on suppose que la variation de Φ est linéaire sur chaque

panneau:

$$\Phi = \Phi_j + \xi(\Phi_{j+1} - \Phi_j)/l_j \tag{25}$$

où Φ_j est le potentiel au noeud j et ξ varie entre 0 et l_j

On remarque de plus que:

$$\Gamma = \Phi_{N+1} - \Phi_1 \tag{26}$$

Après cette discrétisation, l'équation (24) devient :

$$\Phi(P) = \Phi_\infty(P) + \Phi^f(P) - \sum_{j=1}^{N} \int_0^{l_j} \Phi\, \vec{n}\cdot\nabla G(r)\, ds$$

$$- (\Phi_{N+1} - \Phi_1) \int_{S_s} \vec{n}\cdot\nabla G(r)\, ds \tag{27}$$

Après évaluation des intégrales (Annexe 2), on obtient (Fig. 4):

$$\Phi(P) = V_\infty(x(P)\cos\alpha + y(P)\sin\alpha) + \Phi^f(P)$$

$$+ \frac{1}{2\pi} \sum_{j=1}^{N} [\Phi_j + \frac{x^*}{l_j}(\Phi_{j+1} - \Phi_j)]\beta_{P,j} + \frac{y^*}{l_j}(\Phi_{j+1} - \Phi_j) \ln \frac{r_{P,j+1}}{r_{P,j}}$$

$$+ \frac{1}{2\pi}(\Phi_{N+1} - \Phi_1)\beta_{P,N+1} \tag{28}$$

où $x^* = (x(P) - x_j)\cos\theta_j + (y(P) - y_j)\sin\theta_j$ (29)

$$y^* = -(x(P) - x_j)\sin\theta_j + (y(P) - y_j)\cos\theta_j \tag{30}$$

$$r_{P,j} = [(x(P) - x_j)^2 + (y(P) - y_j)^2]^{1/2} \tag{31}$$

$$r_{P,j+1} = [(x(P) - x_{j+1})^2 + (y(P) - y_{j+1})^2]^{1/2} \tag{32}$$

$$\beta_{P,j} = \operatorname{arctg}\left[\frac{(y(P) - y_j)*(x_{j+1} - x(P)) - (y(P) - y_{j+1})*(x_j - x(P))}{(y(P) - y_j)*(y(P) - y_{j+1}) + (x_j - x(P))*(x_{j+1} - x(P))}\right] \tag{33}$$

$$\beta_{P,N+1} = \text{arctg}\left[\frac{y(P)}{c-x(P)}\right] \tag{34}$$

3.1.2 - INTEGRALE DE VOLUME

La discrétisation de l'équation (28) est presque complète. Le dernier terme à discrétiser est $\Phi^f(P)$. Ce terme est égale à:

$$\Phi^f(P) = \int_{\Omega} (M^2 \ \Phi_{ss}) \ G(r) \ d\Omega \tag{34}$$

où Ω est le volume infini entourant le corps.
Le terme $M^2 \ \Phi_{ss}$ n'est important que dans la zone située près du corps et il tend vers zéro en même temps que s'atténue la perturbation due à la présence du corps, c'est-à-dire que $M^2 \ \Phi_{ss}$ devient pratiquement négligeable à partir d'une certaine distance du profil. Dans les problèmes que nous traitons, la région Ω sera donc délimitée par un rectangle centré à mi-corde du profil et dont les dimensions sont finies.

La résolution de l'intégrale (34) est effectuée à l'aide de la méthode des singularités à répartition discrétisée qui permet de remplacer l'intégrale de volume par des sommations d'effets élémentaires de singularité. Pour ce faire, on divise la région Ω en un ensemble de pavés rectangulaires réguliers pénétrant le corps (Fig. 5). On associe à chaque pavé une valeur constante de $M^2 \ \Phi_{ss}$

correspondant à la valeur calculée au centre du pavé considéré. La valeur de $M^2\ \Phi_{ss}$ associée aux pavés possédant leur centre à l'intérieur du corps est égale à zéro.

L'intégrale (34) devient donc

$$\Phi^f(P) = \sum_{n=1}^{Nx} \sum_{m=1}^{Ny} M^2\ \Phi_{ss} \int_{\text{pavé}_{n,m}} G(r)\ d\Omega \tag{35}$$

Après évaluation de l'intégrale (voir Annexe 3), on obtient (Fig. 6):

$$\Phi^f(P) = \frac{1}{4\pi} \sum_{n=1}^{Nx} \sum_{m=1}^{Ny} (M^2\ \Phi_{ss})_{n,m}\ [-d_{x2}d_{y1}\ln R_1^2 + d_{x1}d_{y1}\ln R_2^2$$

$$+ d_{x2}d_{y2}\ln R_3^2 - d_{x1}d_{y2}\ln R_4^2 + d_{y1}^2\theta_1 \tag{36}$$

$$+ d_{x2}^2\theta_2 + d_{y2}^2\theta_3 + d_{x1}^2\theta_4 - 3\Delta x\Delta y]$$

3.1.3 - METHODE DE PANNEAUX CLASSIQUE

Les premières méthodes de panneaux ont été dérivées pour résoudre l'écoulement potentiel incompressible autour de géométries complexes. Pour ce type d'écoulement, le terme de compressibilité $\Phi^f(P)$ de l'équation (28) est égal à zéro. L'équation (28) devient donc:

$$+ \quad \Phi(P) = V_\infty(x(P)\cos\alpha + y(P)\sin\alpha)$$

$$+ \frac{1}{2\pi} \sum_{j=1}^{N} [\Phi_j + \frac{x^*}{l_j}(\Phi_{j+1} - \Phi_j)]\beta_{P,j} + \frac{y^*}{l_j}(\Phi_{j+1} - \Phi_j)\ln \frac{r_{P,j+1}}{r_{P,j}} \tag{37}$$

$$+ \frac{1}{2\pi}(\Phi_{N+1} - \Phi_1)\beta_{P,N+1}$$

Si le point P tend vers le $i^{ième}$ noeud du corps, l'équation (37) possède alors N+1 inconnues, soient les Φ_i, pour i =1,2,3,...,N+1. Cette équation peut devenir un système d'équations algébriques si on l'applique à tous les noeuds (il y a N+1 noeuds). Mais ce n'est pas si simple que ça, on doit faire attention lorsque j = i ou j = i-1 (éqn. (37), P devient i). En effet, ces termes de la sommation contiennent des singularités logarithmiques car $r_{i,i} = 0$. De plus, $\beta_{i,j}$ est indéterminé losque j = i ou j = i-1. Nous allons donc faire tendre le point P sur le $i^{ième}$ noeud et considérer les effets sur l'équation(37). Comme il est dit précédemment, les termes importants sont les $i^{ième}$ et $i-1^{ième}$ de la sommation de l'équation (37). On peut donc écrire:

$$\Phi_i = \frac{1}{2\pi}\left[[\Phi_{i-1} + \frac{x^*_{i-1}}{l_{j-1}}(\Phi_i - \Phi_{i-1})]\beta_{i,i-1} + \frac{y^*_{i-1}}{l_{i-1}}(\Phi_i - \Phi_{i-1})\ln\frac{r_{i,i}}{r_{i,i-1}}\right.$$

$$\left. + \frac{1}{2\pi}[\Phi_i + \frac{x^*_i}{l_j}(\Phi_{i+1} - \Phi_i)]\beta_{i,i} + \frac{y^*_i}{l_i}(\Phi_{i+1} - \Phi_i)\ln\frac{r_{i,i+1}}{r_{i,i}}\right] \quad (38)$$

$$+ \text{ termes réguliers}$$

Dans cette dernière équation, (x^*_{i-1}, y^*_{i-1}) est la position du point P par rapport au panneau i-1. Une remarque semblable s'applique pour (x^*_i, y^*_i). Les termes réguliers sont tous les termes de la sommation de l'équation (37) sauf les termes j = i et j = i-1.

Comme P tend vers le noeud i, on a qu'à poser:

$$x_i^* = 0 \tag{39}$$

$$y_i^* = 0 \tag{40}$$

$$y_{i-1}^* = 0 \tag{41}$$

$$x_{i-1}^* = l_{i-1} \tag{42}$$

Alors l'équation (38) devient:

$$\Phi_i = \frac{1}{2\pi} \Phi_i (\beta_{i,i-1} + \beta_{i,i}) + \text{termes réguliers} \tag{43}$$

où $\beta_{i\,i-1}$ et $\beta_{i\,i}$ sont indeterminés. Par contre leur somme peut être calculée et représente l'angle entre les panneaux i-1 et i. Lorsque i est égale à 1, le résultat est différent,l'équation (38) devient (Fig. 7):

$$\Phi_1 = \frac{-1}{2\pi}[\Phi_1 (\delta - 2\pi) + \Phi_{N+1} \lambda] + \text{termes réguliers} \tag{44}$$

L'application de l'équation (37) aux noeuds 1 et N+1 donne le même résultat. Nous avons donc N équations linéairement indépendantes. Ceux-ci ne sont pas suffisantes pour déterminer les N+1 inconnues Φ_i. La dernière équation nécessaire provient de l'application de la condition de Kutta au bord de fuite [10]. Elle consiste à égaliser les vitesses tangentielles aux centres des deux panneaux adjacents au bord de fuite:

$$\frac{\Phi_{N+1} - \Phi_N}{l_N} = \frac{\Phi_1 - \Phi_2}{l_1} \tag{45}$$

Ce système d'équation peut être écrit sous la forme matricielle suivante:

$$[K]\ \{\Phi\} = -\{\Phi_\infty\} \tag{46}$$

où [K] est une matrice carré (N+1) X (N+1) appelée la matrice des coefficients d'influence. Elle dépend que de la géométrie du corps.

$\{\Phi\}$ est un vecteur de dimension N+1. Il contient les inconnues Φ_i .

$\{\Phi_\infty\}$ est aussi un vecteur de dimension N+1. Il est calculé à l'aide de l'équation (17). Ce vecteur ne dépend que de la géométrie du corps et de l'angle d'attaque.

En résolvant le système (46), il est possible de déterminer directement les N+1 inconnues.

3.1.4 - METHODE DE PANNEAUX TRANSSONIQUE

Pour le cas des écoulements compressibles, le terme Φ^f est différent de zéro et le système matriciel (46) devient:

$$[K]\ \{\Phi\} = -\{\Phi_\infty\} - \{\Phi^f\} \tag{47}$$

où $\{\Phi^f\}$ est un vecteur de dimension N+1. Il est calculé à l'aide de l'équation (36). Ce terme dépend du vecteur solution $\{\Phi\}$.

Une solution directe de ce système est irréalisable puisque que le terme de droite dépend de la solution. C'est

la caractéristique d'un système non-linéaire. Pour le résoudre, on doit procéder par un calcul itératif. La procédure est expliqué à la section 3.2.

3.1.5 - COEFFICIENT DE PRESSION

Nous devons maintenant discrétiser l'équation (14). Cette équation donne une relation entre le coefficient de pression C_p et la vitesse $\vec{V}$ $(\nabla\Phi)$. Pour calculer les coefficients de pression sur le corps, on doit donc connaître les vitesses sur le corps. La solution de l'équation (28) nous permet de connaître les valeurs du potentiel Φ à chaque noeud. Comme la vitesse représente le gradient du potentiel, la procédure la plus simple est de dériver l'équation (25), la formule supposant une variation linéaire du potentiel sur chaque panneau, et de considérer le résultat comme étant la vitesse tangentielle au centre du panneau :

$$V_{t\,i} \approx \frac{\Phi_{i+1} - \Phi_i}{l_i} \tag{48}$$

L'expression pour le coefficient de pression au centre du panneau i devient donc:

$$C_{p\,i} = \frac{2}{\gamma M_\infty^2}\left[\left[\frac{\gamma-1}{2}M_\infty^2(1-V_{t\,i}{}^2)+1\right]^{\frac{\gamma}{\gamma-1}}-1\right] \tag{49}$$

3.1.6 - COEFFICIENT DE PORTANCE

Par définition, le coefficient de portance s'exprime de la façon suivante:

$$C_L = \frac{L}{\frac{1}{2}\rho_\infty V_\infty^2 c} \tag{50}$$

De plus, selon la relation de Kutta-Joukowski

$$L = \rho_\infty V_\infty \Gamma \tag{51}$$

En combinant les équation (50) et (51), on obtient:

$$C_L = 2\frac{\Gamma}{V_\infty c} \tag{52}$$

Enfin, en utilisant la relation(26), on trouve:

$$C_L = 2\frac{\Phi_{N+1} - \Phi_1}{V_\infty c} \tag{53}$$

3.1.7 - DISTRIBUTION DES PANNEAUX

Avec la méthode de panneaux décrit précédemment, le corps est approximé par un ensemble de lignes droites que l'on nomme panneaux. Or, dans les problèmes qui nous intéressent, ce corps est un profil aérodynamique. Pour bien représenter le profil, on doit concentrer les panneaux au bord d'attaque, c'est-à-dire à l'endroit où la courbure est très importante, ainsi que dans le voisinage du bord de fuite afin que la condition de Kutta (équation (45)) soit soit appliquée le plus près possible du bord de fuite. Pour

ce faire, on doit définir une distribution de panneaux. Pour les écoulements sans choc, un choix est la distribution cosinusoïdale. Elle s'exprime de la façon suivante (Fig. 3):

- on calcul un incrément constant $\Delta\theta$ en fonction du nombre de panneaux désiré:

$$\Delta\theta = \frac{2\pi}{N} \tag{54}$$

- on calcul un ensemble de coordonnées x en utilisant les relations suivantes:

$$x_i = \frac{c}{2}(1 + \cos\theta_i) \tag{55}$$

$$\theta_i = i\Delta\theta \quad \text{avec } i = 1,2,3,\ldots,N+1 \tag{56}$$

- les coordonnées y_i correspondantes sont calculées avec une relation de la forme suivante:

$$y_i = f(x_i) \tag{57}$$

Par exemple, pour un NACA 0012, on a [10]:

$$y = -.6(.2969x^{1/2} - .1260x - .3537x^2 + .2843x^3 - .1015x^4) \tag{58}$$

Pour les écoulements avec choc, on doit utiliser une distribution possédant les même caractéristiques que dans le cas sans choc. De plus, pour mieux prédire un choc abrupt, on doit prévoir une concentration de panneaux dans la région du choc. On propose deux distributions cosinusoïdales de part et d'autre du choc. Il s'agit tout d'abord d'estimer la position du choc que l'on note x_{choc}. Alors, les coordonnées x_i sont calculées en utilisant les relations suivantes:

pour $x_i < x_{choc}$

$$x_i = \frac{x_{choc}}{2} (1 + \cos\theta_i) \tag{59}$$

pour $x_i > x_{choc}$

$$x_i = \frac{c - x_{choc}}{2} (1 + \cos\theta_i) \tag{60}$$

Les coordonnées y_i correspondantes sont calculées à l'aide de la relation (57).

3.1.8 - DETERMINATION DES POINTS INTERIEURS

L'évaluation de l'intégrale de volume (35) doit être réalisée en excluant les pavés situés à l'intérieur du corps. La méthode la plus simple pour identifier ces derniers points et qui s'intègre facilement à l'algorithme que nous proposons au paragraphe 3.2 consiste à calculer la circulation autour du corps (ou plus précisément sur le contour S_c) générée par un tourbillon placé au centroïde du pavé à identifier. De cette façon, si la circulation est nulle, on peut dire que le pavé (ou plutôt le centroïde) est situé à l'extérieur du corps.

La circulation autour du corps, générée par un tourbillon ponctuel unitaire au centroïde n,m, est donnée par :

$$CIRC = \int_{S_c} \vec{V} \cdot d\vec{l} \tag{61}$$

où $\vec{V} = \dfrac{\vec{\theta}}{2\pi r}$ (62)

$$CIRC = \sum_{j=1}^{N} \int_{0}^{l_j} \vec{V} \cdot d\vec{l} \qquad (63)$$

Comme on est en présence d'un écoulement potentiel généré par une singularité, l'intégrale que l'on retrouve dans l'équation (63) ne dépend pas du parcours mais uniquement des points de départ et d'arrivée. On peut donc utiliser l'expression suivante (Fig. 8):

$$CIRC = \sum_{j=1}^{N} \int_{S_A} \vec{V} \cdot d\vec{l} + \int_{S_B} \vec{V} \cdot d\vec{l} \qquad (64)$$

où S_A est un parcours radial

S_B est un arc de cercle

Alors, l'intégrale sur S_A est nulle puisque la vitesse $\vec{V}$ est perpendiculaire. De plus, la vitesse est parallèle et la distance r est constante sur S_B, ce qui nous permet d'écrire:

$$CIRC = \frac{1}{2\pi} \sum_{j=1}^{N} \frac{1}{r} \int_{l_j} dl \qquad (65)$$

$$CIRC = \frac{1}{2\pi} \sum_{j=1}^{N} \beta_{P,j} \qquad (66)$$

3.2 - ALGORITHME

La résolution du système d'équations (47) est effectuée suivant un procédé itératif. On considère d'abord l'écoulement comme incompressible, c'est-à-dire qu'on résout le système en admettant que $M^2\ \Phi_{ss}$ est nul dans tout le domaine Ω, ce qui permet d'évaluer en première approximation la dérivée directionnelle du potentiel (Φ_{ss}) et le nombre de Mach au centre de chaque pavé, tant dans la zone subsonique que supersonique.

La dérivée Φ_{ss} est calculée en utilisant le schéma tourné de Jameson [4] pour ainsi capturer automatiquement les chocs caractérisant les écoulements transsoniques.

De cette valeur de Φ_{ss} estimée et du nombre de Mach local, on calcule $\Phi^f(P)$ aux noeuds du corps en utilisant l'équation (36). Le cycle recommence jusqu'à ce que la convergence soit atteinte, c'est-à-dire lorsque les coefficients de pression ne varient presque plus. Voici une description plus complète du procédé itératif:

3.2.1 - PROCEDE ITERATIF

1. On pose $M^2\ \Phi_{ss} = 0$ au centre de tous les pavés.
   On pose également $\{\Phi^f\} = 0$.
2. On solutionne le système (47) par une méthode de panneaux.
3. On calcule Φ^p au centre de tous les pavés.

4. On superpose Φ^p et Φ^f pour obtenir Φ au centre de tous le pavés.
5. On calcule $M^2 \Phi_{ss}$ au centre de chaque pavé en utilisant le schéma tourné.
6. On calcule Φ^f au centre de tous les pavés en utilisant l'équation (36).
7. On calcule $\{\Phi^f\}$, le terme de droite du système (47) en utilisant l'équation (36).

On répète les étapes 2 à 7 jusqu'à la convergence de la solution.

Le maillage rectangulaire utilisé pour obtenir la solution du problème pénètre le corps à analyser. Il n'est pas nécessaire de construire un maillage adapté au corps puisqu'aucune condition aux frontières du corps est appliquée sur les points du maillage pour le calcul du terme $M^2 \Phi_{ss}$. Ceci constitue le grand intérêt de cette méthode.

3.2.2 METHODE DE PANNEAUX AVEC DOUBLETS LINEAIRES

Comme il a été dit dans l'introduction, il existe un nombre impressionnant de méthodes de panneaux. Ces méthodes diffèrent principalement au niveau des singularités utilisées et de leur distribution. La méthode présentée au paragraphe 3.1 soulève sûrement quelques questions concernant le choix de la discrétisation de Φ sur chaque

panneaux (équation (25)). Nous avons choisi une variation linéaire. Plusieurs autres méthodes utilisent plutôt une distribution constante: le potentiel est constant sur chaque panneau mais varie d'un panneau à un autre. Avec un tel choix, les expressions utilisées pour l'évaluation de la matrice d'influence [K] sont grandement simplifiées. Le calcul de la matrice [K] est donc réalisé plus rapidement.

On peut alors s'intéroger sur la pertinence d'une variation linéaire. La première motivation pour le développement de méthodes de panneaux avec variation linéaire fut basée sur la possibilité qu'en utilisant ce type de variation, on pourrait obtenir une solution aussi précise qu'avec la méthode constante tout en utilisant un nombre moins élevé de panneaux. Le temps de calcul utilisé pour l'évaluation des expressions complexes serait compensé par un moins grand nombre de calculs à réaliser. Ceci n'est pas encore vérifié.

Pour bien justifier le choix que nous avons fait, on doit regarder ce que l'on appelle le système équivalent. Nous représentons la solution Φ par, entre autre, une distribution de doublet sur la surface du corps. Or à toute distribution de doublet correspond une distribution équivalente de tourbillon (voir Annexe 4).

Pour une distribution constante de doublet, le système

équivalent est un ensemble de tourbillons ponctuels situés sur chaque noeud et d'intensité égale à la discontinuité dans la distribution de doublet au noeud. De plus, au bord de fuite, les discontinuités dans les distributions de doublet s'annulent et l'intensité du tourbillon à ce point est zéro. Ce système équivalent illustre très bien les résultats erronés que l'on peut obtenir en utilisant la somme des contributions de chaque panneau (semblable à l'équation (27)) pour l'évaluation du potentiel Φ^P au centre de chaque pavé. Le champ de vitesse induit par un tourbillon ponctuel est beaucoup trop singulier près dans son voisinage. L'erreur augmente à mesure que la distance entre le centroïde du pavé considéré et un noeud du corps est petite. Dans une méthode de panneaux subsonique, on n'a pas à évaluer le potentiel au centre des pavés puisque le terme $\{\Phi^f\}$ est nul: une distribution constante peut donc être utilisée. Un autre problème concernant la distribution constante de doublet est l'impossibilité de spécifier une condition de Kutta explicitement [15]. On obtiendrait ainsi une équation de trop. En effet, la condition de Kutta est implicitement satisfaite puisqu'il n'y a pas de tourbillon ponctuel au bord de fuite. Ceci signifie que la vitesse au bord de fuite est finie. Pour le cas non-linéaire, il est hors de question d'utiliser ce type de distribution.

Pour le cas d'une distribution linéaire de doublet, le

système de tourbillon équivalent est une distribution constante de tourbillon d'intensité égale à la pente de la distribution de doublet. Le champ de vitesse induit est beaucoup moins singulier. Un autre avantage concernant la distribution linéaire est d'offir la possibilité d'appliquer l'équation (28) aux noeuds du corps et non pas au centre de chaque panneau: les noeuds sont réellement sur le corps tandis que les points au centre de chaque panneaux ne le sont pas.

Le champs de vitesse induit par un doublet est nettement plus compliqué que celui induit par un tourbillon. Il est donc normal, puisqu'il existe une relation entre une distribution de doublet et de tourbillon, de justifier ce choix. Pour une extension de la méthode en 3-D, il est préférable d'utiliser des doublets car l'utilisation de tourbillon en 3-D est très complexe puisque l'on doit respecter les lois de Helmhotlz afin de générer un écoulement irrotationnel. On doit en autre s'assurer que les lignes de tourbillons s'étendent jusqu'à l'infini ou forment un parcours fermé.

Il existe plusieurs méthodes pour résoudre le système (47) divisées en deux grandes catégories: les méthodes directes et indirectes. Dans la première catégorie, il y a l'élimination Gaussienne et la décomposition LU. Ces méthodes sont dites directes car elles permettent d'obtenir

directement l'inverse de la matrice. Dans le cas des méthodes indirectes, la matrice n'est jamais inversée et la solution est obtenue itérativement. Dans cette catégorie on peut citer l'itération de Gauss-Seidel et le SOR (Successive over-relaxation). En général, une méthode indirecte nous permet d'obtenir plus rapidement la solution car l'inversion d'une matrice est une tâche énorme. Par contre, dans le contexte actuel, une méthode directe est mieux adaptée puisque que l'on inverse une seule fois la matrice avant de débuter les itérations. Ensuite, la solution est rapidement obtenue par un produit matriciel de la matrice inversée et du vecteur de droite. Nous avons donc adopté la décomposition LU car c'est une méthode directe des plus rapides.

3.2.3 - SCHEMA TOURNE

L'algorithme que nous proposons doit résoudre l'équation du potentiel complet (1) en régime transsonique, c'est-à-dire lorsque l'on a des régions de l'écoulement où les vitesses sont subsoniques et d'autres régions supersoniques. La difficulté d'un tel problème vient du fait que l'équation change de type d'une région à une autre. En effet, aux points subsoniques, l'équation (1) est elliptique tandis que dans la région supersonique, elle est hyperbolique. C'est ce que l'on appelle une équation mixte elliptique-hyperbolique. Or la plupart des méthodes

numériques ne peuvent résoudre qu'un type d'équation. Par exemple, la méthode des panneaux classiques solutionne l'équation de Laplace qui est un problème de type elliptique.

Le grand problème du changement de type d'équation est que la zone d'influence n'est pas la même. En plus de tenir compte de cette différence dans les zones d'influences, l'algorithme doit exclure la présence de chocs d'expansion que l'on ne retrouve pas dans la nature.

Afin de respecter la zone d'influence pour chaque type d'équation, Jameson [4] a conçut un schema qui utilise une représentation par différences finies centrées lors de l'évaluation de la dérivée Φ_{ss} aux points subsoniques et une représentation amont dans la région supersonique. En plus de de bien utiliser les régions d'influence de chaque type d'équation, cette représentation introduit une viscosité artificielle de telle sorte que seulement les chocs de compression peuvent être prédits.

En 2-D, la seconde dérivée du potentiel Φ dans le sens du courant est exprimée de la façon suivante:

$$\Phi_{ss} = \frac{1}{|V|^2}(u^2\Phi_{xx} + 2uv\Phi_{xy} + v^2\Phi_{yy}) \qquad (67)$$

Aux points supersoniques, des représentations amonts de Φ_{xx}, Φ_{xy} et Φ_{yy} doivent être utilisées. Dans les régions

subsoniques, tous les termes sont calculés à l'aide d'expressions centrées. On distingue donc cinq cas différents :

Cas 1: point subsonique

A ces points, on utilise une représentation centrée. Φ_{ss} est calculée à l'aide des expressions suivantes:

$$\Phi_{xx} = \frac{\Phi_{n+1,m} - 2\Phi_{n,m} + \Phi_{n-1,m}}{\Delta x^2} \tag{68}$$

$$\Phi_{xy} = \frac{\Phi_{n+1,m+1} - \Phi_{n+1,m-1} - \Phi_{n-1,m+1} + \Phi_{n-1,m-1}}{4\Delta x\Delta y} \tag{69}$$

$$\Phi_{yy} = \frac{\Phi_{n,m+1} - 2\Phi_{n,m} + \Phi_{n,m-1}}{\Delta y^2} \tag{70}$$

Cas 2: point supersonique, $u > 0$ et $v > 0$

A ces points, Φ_{ss} est calculée à l'aide des

$$\Phi_{xx} = \frac{\Phi_{n,m} - 2\Phi_{n-1,m} + \Phi_{n-2,m}}{\Delta x^2} \tag{71}$$

$$\Phi_{xy} = \frac{\Phi_{n,m} - \Phi_{n,m-1} - \Phi_{n-1,m} + \Phi_{n-1,m-1}}{\Delta x\Delta y} \tag{72}$$

$$\Phi_{yy} = \frac{\Phi_{n,m} - 2\Phi_{n,m-1} + \Phi_{n,m-2}}{\Delta y^2} \tag{73}$$

Cas 3: point supersonique, $u > 0$ et $v < 0$

A ces points, Φ_{ss} est calculée à l'aide des équations suivantes:

$$\Phi_{xx} = \frac{\Phi_{n,m} - 2\Phi_{n-1,m} + \Phi_{n-2,m}}{\Delta x^2} \tag{74}$$

$$\Phi_{xy} = \frac{\Phi_{n,m+1} - \Phi_{n,m} - \Phi_{n-1,m+1} + \Phi_{n-1,m}}{\Delta x \Delta y} \tag{75}$$

$$\Phi_{yy} = \frac{\Phi_{n,m} - 2\Phi_{n,m+1} + \Phi_{n,m+2}}{\Delta y^2} \tag{76}$$

<u>Cas 4: point supersonique, u < 0 et v > 0</u>

A ces points, Φ_{ss} est calculée à l'aide des équations suivantes:

$$\Phi_{xx} = \frac{\Phi_{n,m} - 2\Phi_{n+1,m} + \Phi_{n+2,m}}{\Delta x^2} \tag{77}$$

$$\Phi_{xy} = \frac{\Phi_{n+1,m} - \Phi_{n+1,m-1} - \Phi_{n,m} + \Phi_{n,m-1}}{\Delta x \Delta y} \tag{78}$$

$$\Phi_{yy} = \frac{\Phi_{n,m} - 2\Phi_{n,m-1} + \Phi_{n,m-2}}{\Delta y^2} \tag{79}$$

<u>Cas 5: point supersonique, u < 0 et v < 0</u>

A ces points, Φ_{ss} est calculée à l'aide des équations suivantes:

$$\Phi_{xx} = \frac{\Phi_{n,m} - 2\Phi_{n+1,m} + \Phi_{n+2,m}}{\Delta x^2} \tag{80}$$

$$\Phi_{xy} = \frac{\Phi_{n+1,m+1} - \Phi_{n+1,m} - \Phi_{n,m+1} + \Phi_{n,m}}{\Delta x \Delta y} \tag{81}$$

$$\Phi_{yy} = \frac{\Phi_{n,m} - 2\Phi_{n,m+1} + \Phi_{n,m+2}}{\Delta y^2} \tag{82}$$

La valeur de Φ_{ss} dépend de u et v qui à leur tour dépendent de Φ_{ss}. Pour calculer Φ_{ss} correctement, on doit répéter les étapes 4,5 et 6 plusieurs fois avant de passer à l'étape 7 (voir paragraphe 3.2.1). Cette boucle interne est réalisée environ cinq fois. Ce nombre peut paraître petit si on le compare avec le nombre d'itérations nécessaires pour la convergence du schéma tourné de Jameson appliqué dans la méthode des différences finies. Dans ce type de solution, la plupart des itérations sont nécessaires pour introduire les effets des conditions aux frontières partout dans le domaine de calcul. Le cadre dans lequel on utilise ce schéma est complètement différent de celui des différences finies. En effet, les étapes 2,3 et 4 nous permettent d'obtenir une solution qui respecte les conditions aux frontières et la boucle interne ne sert qu'à prendre en compte de la dépendance entre Φ_{ss}, u et v (i.e. la non-linéarité).

3.3 - DESCRIPTION DU PROGRAMME

Un code utilisant l'algorithme décrit au paragraphe 3.2 précédent à été réalisé en FORTRAN 77. Le programme a été implanté sur un ordinateur central puisqu'il utilise de grandes ressources autant au niveau de la mémoire que du temps de calcul. Le listing est en annexe 5.

3.3.1 - DONNEES

Les données nécessaires à l'exécution du programme sont lues dans un fichier par la sous-routinne INPUT. Les principales données sont:

- nom du corps : AIRFOIL
- type de corps : si IGEOM = 1, le programme construit un profil NACA00xx.
  si IGEOM = 2, le programme construit un cercle.
- variable logique DEBUG : si DEBUG = .T., on imprime les détails de l'écoulement à chaque itération.
  si DEBUG = .F., on imprime les détails de l'écoulement à la fin seulement.
- variable logique GRAPH : si GRAPH = .T., on crée des fichiers pour production de graphiques.
  si GRAPH = .F., on crée aucun fichier.
- précision désirée : PRECP
- nombre d'itérations maximum : ITMAX
- épaisseur relative du profil : T
- nombre de panneaux avant le choc : N1
- nombre de panneaux après le choc : N2
- nombre de pavés en x : NX

- nombre de pavés en y : NY
- position du choc : XM
- longueur du domaine en x, m : LX
- longueur du domaine en y, m : LY
- angle d'attaque, deg. : ALPHA
- nombre de Mach : MACH
- rapport des chaleurs spécifiques : GAMMA

Les coordonnées x_i et y_i du corps sont automatiquement générées par le code.

3.3.2 - RESULTATS

Les résultats sont envoyés à l'imprimante à l'aide des sous-routines RESULT et IMPR2. On y retrouve les coefficients de pression sur le corps, le coefficient de portance, les vitesses et le nombre de Mach au centre des pavés. Tous ces résultats sont sauvegardés sur fichiers pour la production des graphiques à l'aide de la sous-routine WRESULT.

3.3.3 - SOUS-ROUTINES

Voici la liste des sous-routines avec leur fonction:

- APVT : Cette sous-routine trouve le plus grand pivot d'une matrice. Elle est utilisée lors de l'inversion de la matrice [K].
- CALPF : Cette sous-routine calcule le potentiel Φ^f au centre des pavés à l'aide de l'équation (36).

- CALPHF : Cette sous-routine extrapole les valeurs des potentiels Φ^f aux points à l'intérieur du corps.
- CALPHP : Cette sous-routine calcule le potentiel Φ^P au centre de tous les pavés à l'aide de l'équation (28).
- CIRCLE : La génération des coordonnées y_i d'un cercle est réalisée par cette sous-routine.
- COEF : Cette sous-routine calcule:
  - la matrice [K];
  - les coefficients de l'équation (36) où le point P devient le centre de chaque panneau et le centroïde de chaque pavé;
  - les coefficients de l'équation (28) où le point P devient le centroïde de chaque pavé.

  Elle détermine également les points intérieurs en utilisant la méthode décrite au paragraphe 3.1.8.
- COSIN : Calcul de l'inclinaison θ_i de chaque panneau.
- DIFCOR : Cette sous-routine calcule les dérivées du potentiel Φ^P pour obtenir $\partial\Phi^P/\partial x$, $\partial\Phi^P/\partial y$, $\partial^2\Phi^P/\partial x^2$, $\partial^2\Phi^P/\partial x\partial y$.
- DIFF : Cette sous-routine calcule les dérivées du potentiel Φ^f pour obtenir $\partial\Phi^f/\partial x$, $\partial\Phi^f/\partial y$.
- FINF : Elle calcule Φ^f au centre de chaque panneau.
- GRID : Cette sous-routine calcule les dimensions des

pavés ainsi que la position de leur centre.

- IMPR : Cette sous-routine imprime les détails de l'écoulement telle que le champs de vitesses et les nombres de Mach local.
- INPUT : C'est la sous-routine de lecture des données.
- LONG : La longueur de chaque panneau est calculée dans cette sous-routine.
- LUDCMP : Cette sous-routine exécute la décomposition LU de la matrice [K].
- NACA : Cette sous-routine génère les coordonnées adimensionnelles y_i d'un profil NACA 00xx.
- PTMIL : Le point milieu de chaque panneau est calculé dans cette sous-routine.
- RESULT : Cette sous-routine imprime les coefficients de pression sur le corps ainsi que le coefficient de portance.
- SOLVLU : Cette sous-routine trouve la solution du système (47) en utilisant la décomposition LU de la matrice [K].
- SOURCE : On retrouve dans cette sous-routine le schéma de Jameson. Elle calcule le terme $M^2\Phi_{ss}$ au centre des pavés.
- SUPERP : Cette sous-routine superpose Φ^p et Φ^f pour obtenir le potentiel total de l'écoulement Φ.
- VITPORT: Cette sous-routine calcule les vitesses tangentielles au centre des panneaux et le

coefficient de portance.

- WRESULT: Cette sous-routine crée les fichiers de résultats nécessaires pour la réalisation de graphiques.

CHAPITRE 4

DISCUSSION DES RESULTATS

Les résultats que nous présentons ont été obtenus pour l'écoulement compressible autour d'un profil NACA 0012 pour des nombres de Mach entre 0.7 et 0.8 et des angles d'attaque jusqu'à 1°. Nous avons choisi ce profil car c'est la configuration utilisée par la plupart des auteurs oeuvrant dans le domaine des écoulements transsoniques. De cette façon, il est facile de faire des comparaisons avec des méthodes existantes et des résultats expérimentaux.

Nous avons choisi de comparer nos résultats avec le code éprouvé BGK [16]. Ce programme solutionne l'équation du potentiel complet en utilisant les différences finies. Le profil est transformé en un cercle et le domaine infini entourant ce cercle est ensuite transformé en un domaine fini à l'intérieur du cercle. De cette façon, la condition aux frontières à l'infini est facilement appliquée dans le domaine transformé au centre du cercle. Cette méthode est très précise et nous donnera donc une bonne indication sur la validité de notre méthode. De plus, le code BGK utilise le même schéma rotatif pour capturer les chocs. Ceci est extrêmement important puisque des différences notables sont rencontrées sur la position et la puissance du choc selon que l'on utilise un schéma ou un autre. Or dans la comparaison qui va suivre, on veut analyser surtout la

validité d'une solution intégrale par rapport à une solution par différence finie sans égard au schéma qui capture les chocs.

Les résultats expérimentaux utilisés dans la comparaison qui va suivre ont été obtenus dans le tunnel S_3MA de l'ONERA [17]. Les nombres de Reynolds sont de l'ordre de 10^6, ce qui est considéré comme modérément élevé. La transition a été laissée libre: aucun dispositif a été installé pour forcer la transition à s'opérer à un endroit précis.De plus, aucune correction a été appliquée concernant les effets de parois.

Les résultats numériques que nous présentons portent sur trois conditions distinctes. Il y a les conditions sous-critiques et critiques avec angle d'attaque nul et finalement la condition critique avec angle d'attaque non nul.

La Fig. 9 illustre une comparaison des coefficients de pression entre la methode proposée dans ce travail, le code BGK et les résultats expérimentaux pour le régime sous-critique avec angle d'attaque nul, c'est-à-dire que le nombre de Mach amont est élevé (0.703) mais aucune région supersonique existe autour du profil. Pour ce regime, on peut voir que les deux méthodes sont similaires et se comparent favorablement aux résultats expérimentaux. On

remarque toutefois que BGK surestime légèrement le coefficient de pression près du bord d'attaque, c'est-à-dire où l'on atteint la dépression maximale. Ceci s'explique du fait que l'on utilise l'hypothèse d'un fluide non-visqueux. La Fig. 10 nous montre la distribution du nombre de Mach local dans le domaine de calcul. On remarque qu'il n'existe effectivement aucune région supersonique, mais que les nombres de Mach local sont élevés à certains endroits. On retrouve les nombres de Mach les plus élevés à environ $x/c = 0.3$. Les résultats (Distribution de c_p et nombres de Mach local) pour le régime critique (Mach amont de 0.803) à un angle d'attaque nul sont présentés aux Figs. 11 et 12. Cette fois-ci, on est en présence d'un choc (Fig. 11). BGK surestime toujours le coefficient de pression mais son choc est abrupt et situé à une distance raisonnable du choc expérimental. Par contre son intensité (discontinuité) est trop faible. La méthode intégrale se compare très bien avec les résultats expérimentaux pour ce qui a trait à la prédiction de la pression. Toutefois, le choc est moins bien défini. Ceci indique une limite de la méthode intégrale. Kandil et al. [18] travail sur une technique permettant aux méthodes intégrales de prédire un choc plus abrupt. Elle consiste à l'introduction de panneaux le long du choc. La Fig. 12 indique la position des poches supersoniques. On remarque qu'elles sont de petite dimension. On peut voir également une concentration des

lignes isomachs près de $x/c = 0.5$. C'est la position du choc.

Les Figs. 13 et 14 ont été obtenues pour le régime critique avec angle d'attaque non-nul, soit un nombre de Mach amont de 0.754 et un angle de 0.99. Les mêmes remarques s'appliquent pour ce régime. En effet, à la Fig. 13, on voit que la méthode BGK prévoit des coefficients de pression légèrement trop élevés en amont du choc tandis que le choc calculé par la méthode intégrale est très dissipé. La Fig. 14 illustre la dissymétrie de la solution par rapport à la corde ainsi que la position de l'unique région supersonique sur le dessus du profil.

La méthode proposée résout l'équation du potentiel complet. Cette équation est obtenue en utilisant la condition d'irrotationnalité. Cette condition signifie essentiellement que l'écoulement est isentropique limitant les applications aux chocs faibles puisque la présence de chocs fait varier l'entropie. Si on ne pose pas la condition d'irrotationnalité, on obtient les équations d'Euler. Une méthode numérique basée sur la résolution de ces équations peut prédire des chocs forts, c'est-à-dire où l'écoulement est rotationnel. Fang [19] a solutionné ces équations autour d'un profil en utilisant les éléments finis. Une comparaison avec notre méthode intégrale a été réalisée pour les cas sous-critique et critique. Pour le

premier cas, Fig. 15, on remarque aucune différence significative ce qui indique que l'hypothèse d'irrotationnalité pour les écoulements sans choc est justifiée pour de faibles angles d'attaques. Par contre, dans le cas critique (Fig. 16), on voit que la solution obtenue en résolvant les équations d'Euler sont en meilleur accord avec les résultats expérimentaux: la forme de la courbe des coefficients de pression et la puissance du choc sont semblables aux résultats expérimentaux. Pour ce nombre de Mach amont (0.803), le choc est faible et on peut dire que l'écoulement est isentropique et donc irrotationnel. La différence provient plutôt au niveau de l'algorithme de capture des chocs. En effet, l'algorithme utilisé dans la méthode intégrale est non-conservatif puisque nous solutionnons l'équation du potentiel complet sous une forme quasi-linéaire. Les équations d'Euler se présente sous forme conservative et donc les algorithmes utilisant ces équations sont conservatifs. Or, il a déjà été démontré que les schémas conservatifs sont les seuls schémas traitant correctement les chocs [20]. On conclut qu'il est souhaitable d'utiliser un schéma conservatif. Dans cette optique, Bauer et al. [16] ont dérivé un schéma que l'on nomme quasi-conservatif, c'est-à-dire qu'il présente le même comportement que les schémas conservatifs mais est basé sur la solution de l'équation du potentiel complet sous la forme quasi-linéaire. La différence entre le schéma

non-conservatif et le quasi-conservatif est illustrée à la Fig. 17. On remarque que le choc prédit par le schéma quasi-conservatif est plus fort et que la forme de la courbe est similaire aux résultats expérimentaux.

Une brève discussion s'impose au niveau du temps de calcul des différentes méthodes présentées dans ce travail. Pour le type de problème analysé, le temps de calcul utilisé par la méthode intégrale est de 15 min. CPU tandis que BGK nécessite que 1 min. CPU. Le temps de calcul pour obtenir la solution des équations d'Euler par la méthode présenté dans [19] est du même ordre que la méthode intégrale.

CHAPITRE 5

CONCLUSION

La méthode de panneaux transsonique présente des résultats similaires à une solution par différence finie éprouvée. Le grand avantage de la méthode de panneaux sur cette dernière réside en la possibilité de résoudre l'écoulement autour de géométrie complexe. En effet, la méthode par différence finie décrite brièvement dans la discussion des résultats est limitée aux écoulements 2-D autour d'un profil isolé. D'autres méthodes - utilisant les volumes ou éléments finies et solutionnant le problème dans le domaine réel - ne sont pas limitées à ce type d'écoulement mais nécessitent la construction d'un maillage adapté autour de la géométrie. Ceci constitue une tâche extrêmement difficile et délicate, tout particulièrement en 3-D. De son côté, la méthode proposée utilise un maillage rectangulaire pénétrant la configuration. Ceci est réalisable avec une extrême facilité peu importe la complexité de la géométrie à analyser.

Le calcul sur une géométrie plus complexe n'a pas été encore réalisé. Par contre, l'expérience acquise lors du calcul sur un profil NACA 0012 nous permet de dire qu'il serait facile de traiter une géométrie plus complexe comme par exemple, un profil avec plusieurs volets. La solution d'un tel problème ne nécessiterait aucun changement au

niveau de l'algorithme mais exigerait beaucoup plus de temps de calcul. En effet, ceci augmenterait les dimensions de la matrice d'influence [K] que l'on doit inverser.

La méthode intégrale est donc très avantageuse et l'extension aux écoulements 3-D réalisable. Par contre, une étude encore plus approfondie au niveau du code 2-D est bien nécessaire. En effet, l'algorithme proposé exige une ressource importante au niveau de la mémoire et du temps de calcul. L'utilisation de pavés de même dimension nous a permis de diminuer d'une façon significative l'espace mémoire nécessaire. Par contre, ceci augmente le temps de calcul car on utilise de petits pavés partout dans le domaine alors qu'ils sont nécessaires seulement près du bord d'attaque et du choc. Avant de passer à la réalisation du code 3-D, on doit donc se pencher sur le problème du temps de calcul. Une vectorisation du code nous permettrait facilement de réduire le temps de calcul. Cette vectorisation n'exigera que très peu de changements au niveau de l'algorithme. Les changements résideront surtout au niveau de la programmation. En faisant tourner le programme sans modification sur le IBM 3090 avec l'option vectorisation, on a obtenu un temps de calcul 3 fois moins élevé que sans l'option vectorisation. Une autre solution qui devra être considérée est l'utilisation des différences finies pour le calcul du potentiel Φ^f. On pourrait obtenir

le potentiel Φ^f au centre des pavés en solutionnant une équation de Poisson où le terme de source serait égal à $M^2 \Phi_{ss}$. Ce problème devrait théoriquement être résolu sur un domaine infini. Par contre, l'utilisation d'un algorithme de frontières rapprochées [21] pourrait réduire ce domaine en une région près du corps. Ceci permettrait surement de diminuer et l'espace mémoire et le temps de calcul: la solution d'une équation de Poisson sur un domaine rectangulaire peut être réalisée rapidement.

Finalement, on doit améliorer la prédiction du choc. Pour ce faire, il est recommandé d'utiliser le schéma quasi-conservatif lors de l'évaluation des écoulements critiques. Une étude concernant l'utilisation d'un tel schéma dans l'environnement différent des méthodes de panneaux transsoniques est donc souhaitable. La technique de Kandil et al. [18] ajoute de la complexité et du temps de calcul à l'algorithme mais devrait être envisagé si l'on veut obtenir un choc abrupt.

REFERENCES

[1] Sytsma, H.S., Hewitt, B.L., Rubbert, P.E., "A Comparison of Panel Methods for Subsonic Flow Computation", AGARDograph 241, 1979.

[2] Jameson, A., "Successes and Challenges in Computational Aerodynamics", AIAA 8th Computational Fluid Dynamics Conference, June 1987, Hawaii.

[3] Murman, E.M., Cole, J.D., "Calculation of Plane Steady Transonic Flows", AIAA Journal, 1971, pp. 114-121.

[4] Jameson, A., "Iterative Solution of Transonic Flows over Airfoils and Wings, Including Flows at Mach 1", Comm. Pure Appl. Math., Vol. 27, 1974, pp. 283-309.

[5] Johnson, J.T., James, R.M., Bussoletti, J.E., Woo, A.C., "A Transonic Rectangular Grid Embedded Panel Method", AIAA paper 82-0953, 1982.

[6] Sinclair, P.M., "An Exact Integral (Field Panel) Method for the Calculation of two-Dimensional Transonic Potential Flow around Complex Configurations", Aeronautical J., June/july 1986, pp. 227-236.

[7] Erickson, L.L., Madson, M.D., Woo, A.C., "Apllication of the TranAir Full-Potential Code to the F-16A", Journal of Aircraft, Aug. 1987, Vol.

24, No. 8, p. 540-545.

[8] Sinclair, P.M., "A Three-Dimensional Field-Integral Method for the Calculation of Transonic Flow on Complex Configurations - Theory and Preliminary Result", Aeronautical J., June/July 1988, pp. 235-241.

[9] Anderson, D.A., Tannehill, J.C., Pletcher, R.H., "Computational Fluid Mechanics and Heat Transfer", Hemisphere Publishing Corp., 1984, 599 p.

[10] Moran, J., "Theoretical and Computaional Aerodynamics", John Wiley & Sons, 1984, 464 p.

[11] Mavriplis, F., "Aerodynamic Research on High Lift System", Canadian Aeronautics and Space Journal, May 1971, pp. 175-183.

[12] Hess, J.L., Smith, A.M.O., "Calculation of Nonlifting Potential Flow about Arbritary Three-Dimensional Bodies", ES40622, Douglas Aircraft Co., 1962.

[13] Carmichael, R.L., Erickson, L.L., "PAN AIR - A Higher Order Panel Method for Predicting Subsonic or Supersonic Linear Potential Flows about Arbitrary Configurations", AIAA Paper 81-1255, 1981.

[14] Oskam, B., "Transonic Panel Method for the Full Potential Equation Applied to Multicomponent Airfoils", AIAA Journal, Vol. 23, No. 9, Sept.

1985, pp. 1327-1334.

[15] Freedman, M.I., "Introduction to the Green's Function Method in Aerodynamics", Computational Methods in Potential Aerodynamics, Edited by L. Morino, Springer-Verlag.

[16] Bauer, F., Garabedian, P., Korn, D., "Supercritical Wing Sections", Lecture Notes in Economics and Mathematics Systems, Vol. I & II, Springer-Verlag, Berlin, 1975.

[17] Thibert, J.J., Grandjacques, M., Ohman, L.H., "NACA 0012 Airfoil, An Experimental Data Base for Computer Program Assessment", AGARD-AR-138, May 1979, pp. A1.1-A1.19.

[18] Kandil, O.A., Hong, H., "Full-Potential Integral Solution for Transonic Flows with and without Embedded Euler Domains", AIAA Journal, Sept. 1988, Vol. 26, No. 9, pp. 1079-1086

[19] Fang, Z., Saber, A.J., "A Stream Function Vorticity Pressure Potential formulation for Solution of the Steady Euler Equation", Canadian Aeronautics and Space Journal, Dec. 1988, Vol. 34, No. 4, pp. 239-247.

[20] Jameson, A., "Numerical Computation of Transonic Flows with Shocs Waves", Symposium Transsonicum II, Gottingen, Sept. 1975, pp. 384-414.

[21] Niederdrenk, P., Wedemeyer, E., "Analytic Near-Field Boundary Conditions for Transonic Flow Computations", AIAA Journal, June 1987, Vol. 25, No. 6, pp.884-886.

ANNEXE 1

EQUATION DU POTENTIEL COMPLET

Si l'on désigne par ρ la masse spécifique du fluide, par p sa pression et $\vec{V}$ sa vitesse, les équations de la continuité et du mouvement d'un écoulement non-visqueux s'écrivent respectivement:

$$\nabla \cdot \vec{V} + \frac{\vec{V}}{\rho} \cdot \nabla\rho = 0 \tag{83}$$

$$(\vec{V} \cdot \nabla)\vec{V} = -\frac{\nabla p}{\rho} \tag{84}$$

Si le mouvement est irrotationnel et isentropique, il existe une relation entre p et ρ:

$$\frac{p}{\rho^{\gamma}} = cte \tag{85}$$

De plus, par définition:

$$a^2 = \frac{dp}{d\rho} \tag{86}$$

De ces deux dernières relations, on obtient:

$$a^2 = \frac{\gamma p}{\rho} \tag{87}$$

où a désigne la vitesse du son. De plus:

$$(\vec{V} \cdot \nabla)\vec{V} = \nabla\frac{V^2}{2} + \vec{V} \times (\nabla \times \vec{V}) = \nabla\frac{V^2}{2} \tag{88}$$

puisque l'écoulement est irrotationnel ($\nabla \times \vec{V} = 0$). Les

équations (83) et (84) deviennent:

$$\nabla\cdot\vec{V} + \frac{\vec{V}}{a^2}\cdot\frac{\nabla p}{\rho} = 0 \tag{89}$$

$$\nabla\frac{|\vec{V}|^2}{2} = -\frac{\nabla p}{\rho} \tag{90}$$

Du fait que l'écoulement est irrotationnel, la vitesse $\vec{V}$ dérive d'un potentiel Φ avec $\vec{V} = \nabla\Phi$. On obtient finalement l'équation du potentiel complet en éliminant $\nabla p/\rho$ des relations (89) et (90):

$$\nabla^2\Phi = \frac{1}{2a^2}\nabla\Phi\cdot\nabla|\nabla\Phi|^2 \tag{91}$$

En utilisant les équations (86) et (87), on peut réécrire l'équation (90) sous la forme suivante:

$$\nabla\left[\frac{1}{2}|\vec{V}|^2 + \frac{\gamma}{\gamma-1}\frac{p}{\rho}\right] = 0 \tag{92}$$

En intégrant cette dernière équation, on obtient l'équation de Bernoulli-Lagrange :

$$\frac{1}{2}|\vec{V}|^2 + \frac{\gamma}{\gamma-1}\frac{p}{\rho} = \text{constante} \tag{93}$$

Ou encore

$$\frac{1}{2}|\nabla\Phi|^2 + \frac{\gamma}{\gamma-1}\frac{p}{\rho} = \frac{1}{2}|\vec{V}_\infty|^2 + \frac{\gamma}{\gamma-1}\frac{p_\infty}{\rho_\infty} \tag{94}$$

En utilisant les relations (87) et (94), on obtient:

$$a^2 = \frac{1}{V_\infty^2}\left[\frac{\gamma-1}{2}\left[1 - \frac{V^2}{V_\infty^2}\right] + \frac{1}{M_\infty^2}\right] \tag{95}$$

En posant $V_\infty = 1$, l'équation (~~84~~ 95) devient:

$$a^2 = \frac{1}{M_\infty^2} + \frac{\gamma-1}{2}(1 - V^2) \tag{96}$$

Le coefficient de pression est défini comme suit:

$$C_p = \frac{1}{\cancel{2}}\,\frac{p-p_\infty}{\frac{\rho_\infty U_\infty^2}{2}} \tag{97}$$

En utilisant l'expression de $(p-p_\infty)$ tirée de l'équation de Bernoulli, la relation des gaz parfaits et l'équation (87), on obtient:

$$C_p = \frac{2}{\gamma M_\infty^2}\left[\left[\frac{\gamma-1}{2}M_\infty^2(1-|\nabla\Phi|^2)+1\right]^{\frac{\gamma}{\gamma-1}}-1\right] \tag{98}$$

ANNEXE 2

INTEGRALES DE SURFACE

Cette annexe décrit l'évaluation des intégrales de surface que l'on retrouve dans l'équation (27). La première intégrale à évaluer est la suivante (Fig. 4):

$$IS_1 = \int_0^{l_j} \Phi \, \vec{n}_j \cdot \nabla G(r) \, ds \tag{99}$$

où $\Phi = \Phi_j + \xi(\Phi_{j+1} - \Phi_j)/l_j$ (100)

$$G(r) = \frac{1}{2\pi} \ln(r) \tag{101}$$

De plus,

$$\nabla[\ln(r)] = \frac{1}{r} \nabla r = \frac{\vec{r}}{r^2} \tag{102}$$

$$\vec{n}_j \cdot \nabla[\ln(r)] = \frac{1}{r^2} \vec{n}_j \cdot \vec{r} = \frac{-y^*}{r^2} \tag{103}$$

$$r^2 = (x^* - \xi)^2 + y^{*2} \tag{104}$$

Alors

$$IS_1 = \frac{-\Phi_j}{2\pi} \int_0^{l_j} \frac{y^*}{(x^* - \xi)^2 + y^{*2}} d\xi - \frac{\Phi_{j+1} - \Phi_j}{2\pi l_j} \int_0^{l_j} \frac{y^* \xi}{(x^* - \xi)^2 + y^{*2}} d\xi \tag{105}$$

$$IS_1 = \frac{\Phi_j (x^* - l_j) - \Phi_{j+1} x^*}{2\pi l_j} \int_0^{l_j} \frac{y^* \, d\xi}{(x^* - \xi)^2 + y^{*2}} + \frac{\Phi_{j+1} - \Phi_j}{2\pi l_j} \int_0^{l_j} \frac{y^* (x^* - \xi)}{(x^* - \xi)^2 + y^{*2}} d\xi \tag{106}$$

$$IS_1 = \frac{\Phi_j (x^* - l_j) - \Phi_{j+1} x^*}{2\pi l_j} \operatorname{arctg}\left[\frac{y^*}{x^* - \xi}\right]_0^{l_j} - \frac{\Phi_{j+1} - \Phi_j}{2\pi}\left[\frac{y^*}{l_j} \ln[(x^* - \xi)^2 + y^{*2}]\right]_0^{l_j} \tag{107}$$

$$IS_1 = -\frac{1}{2\pi}\left[\Phi_j + \frac{x^*}{l_j}(\Phi_{j+1} - \Phi_j)\right]\beta_{P,j} - \frac{\Phi_{j+1} - \Phi_j}{2\pi}\frac{y^*}{l_j}\ln\left[\frac{r_{P,j+1}}{r_{P,j}}\right] \tag{108}$$

La seconde intégrale est la suivante:

$$IS_2 = \int_{S_s} \vec{n}_j \cdot \nabla G(r)\, ds \tag{109}$$

$$IS_2 = \frac{-1}{2\pi}\int_C^{\infty} \frac{y}{(\xi - x)^2 + y^2}\, d\xi \tag{110}$$

$$IS_2 = \frac{-1}{2\pi} \operatorname{arctg}\left[\frac{y}{\xi - x}\right]_C^{\infty} \tag{111}$$

$$IS_2 = \frac{-\beta_{P,N+1}}{2\pi} \tag{112}$$

ANNEXE 3

INTEGRALE DE VOLUME

Cette annexe décrit l'évaluation de l'intégrale de volume que l'on retrouve dans l'équation (35):

$$IV = \int_{\text{pavé}_{n,m}} G(r)\ d\Omega \tag{113}$$

$$\text{où } G(r) = \frac{1}{2\pi} \ln(r) \tag{114}$$

Pour l'évaluation de cette intégrale, on doit distinguer deux cas différents: premièrement lorsque le point P est à l'extérieur du pavé et deuxièment lorsqu'il est dans le pavé. Dans ce dernier cas, l'évaluation de l'intégrale doit se faire par un calcul à la limite puisque r tend vers zéro.

Cas 1: P est à l'extérieur du pavé (Fig. 6)

Selon les paramètres apparaissant sur la Fig. 6 et en utilisant la définition de G(r), l'équation (1) devient:

$$IV = \frac{1}{2\pi} \int_{-\delta x}^{\delta x} \int_{-\delta y}^{\delta y} \ln[(x-x^*)^2+(y-y^*)^2]^{1/2}\ dy\ dx \tag{115}$$

$$= \frac{1}{4\pi} \int_{-\delta x}^{\delta x} \left[\int_{-\delta y}^{\delta y} \ln[(x-x^*)^2+(y-y^*)^2]\ dy \right] dx \tag{116}$$

$$= \frac{1}{4\pi} \int_{-\delta x}^{\delta x} \left[(y-y^*) \ln[(x-x^*)^2 + (y-y^*)^2] - 2y \right.$$

$$\left. + 2(x-x^*) \operatorname{arctg}\left[\frac{y-y^*}{x-x^*}\right] \right]_{-\delta y}^{\delta y} dx \tag{117}$$

$$IV = \frac{1}{4\pi} \left[-d_{y2} \int_{-\delta x}^{\delta x} \ln[(x-x^*)^2 + d_{y2}^2] \, dx \right.$$

$$+ d_{y1} \int_{-\delta x}^{\delta x} \ln[(x-x^*)^2 + d_{y1}^2] \, dx$$

$$+ 2 \int_{-\delta x}^{\delta x} (x-x^*) \operatorname{arctg}\left[\frac{-d_{y2}}{x-x^*}\right] dx \tag{118}$$

$$\left. - 2 \int_{-\delta x}^{\delta x} (x-x^*) \operatorname{arctg}\left[\frac{-d_{y1}}{x-x^*}\right] dx - 4\delta y \int_{-\delta x}^{\delta x} dx \right]$$

or

$$I_i = \int_{-\delta x}^{\delta x} \ln[(x-x^*)^2 + d_{y2}^2] \, dx \tag{119}$$

$$I_i = \left[(x-x^*)\ln[(x-x^*)^2+d_{y2}^2] - 2x - 2d_{y2}\ \mathrm{arctg}\left[\frac{x-x^*}{-d_{y2}}\right]\right]_{-\delta x}^{\delta x} \tag{120}$$

$$=-d_{x2}\ln R_3^2+d_{x1}\ln R_4^2-2d_{y2}\left[\mathrm{arctg}\left[\frac{-d_{x2}}{-d_{y2}}\right]-\mathrm{arctg}\left[\frac{-d_{x1}}{-d_{y2}}\right]\right]-4\delta x \tag{121}$$

$$=-d_{x2}\ln R_3^2 + d_{x1}\ln R_4^2 - 2d_{y2}\theta_3 - 4\delta x \tag{122}$$

$$I_{ii} = \int_{-\delta x}^{\delta x} \ln[(x-x^*)^2+d_{y1}^2]\ dx \tag{123}$$

$$=-d_{x2}\ln R_1^2 + d_{x1}\ln R_2^2 + 2d_{y1}\theta_1 - 4\delta x \tag{124}$$

$$I_{iii} = \int_{-\delta x}^{\delta x} (x-x^*)\ \mathrm{arctg}\left[\frac{-d_{y2}}{x-x^*}\right]\ dx \tag{125}$$

posons $f = (x-x^*)^{-1}$ (126)

alors $df = -(x-x^*)^{-2}\ dx = -f^2\ dx$ (127)

$$I_{iii} = \int_{f(-\delta x)}^{f(\delta x)} -f^{-3}\ \mathrm{arctg}\left[-d_{y2}\ f\right]\ df \tag{128}$$

En effectuant par partie, où $u = \mathrm{arctg}[-d_{y2}\ f]$ (129)

$$dv = -f^{-3}\ df \tag{130}$$

$$du = \frac{-d_{y2}}{1 + d_{y2}^{\ 2} f^2}\ df \tag{131}$$

$$v = 0.5\ f^{-2} \tag{132}$$

$$I_{iii} = 0.5\ f^{-2}\,\mathrm{arctg}\left[-d_{y2}\ f\right]\Bigg|_{f(-\delta x)}^{f(\delta x)} + 0.5 \int_{f(-\delta x)}^{f(\delta x)} \frac{d_{y2}\ f^{-2}}{1 + d_{y2}^{\ 2} f^2}\ df \tag{133}$$

$$I_{iii} = 0.5\left[f^{-2}\,\mathrm{arctg}\left[-d_{y2} f\right] - d_{y2} f^{-1} + d_{y2}^2\,\mathrm{arctg}\left[-d_{y2} f\right]\right]_{f(-\delta x)}^{f(\delta x)} \tag{134}$$

$$I_{iii} = 0.5\left[\ d_{x2}^2\,\mathrm{arctg}\left[\frac{-d_{y2}}{-d_{x2}}\right] + d_{x2} d_{y2} + d_{y2}^2\,\mathrm{arctg}\left[\frac{-d_{y2}}{-d_{x2}}\right]\right.$$
$$\left. -\ d_{x1}^2\,\mathrm{arctg}\left[\frac{-d_{y2}}{-d_{x1}}\right] - d_{x1} d_{y2} - d_{y2}^2\,\mathrm{arctg}\left[\frac{-d_{y2}}{-d_{x1}}\right]\right] \tag{135}$$

$$I_{iv} = \int_{-\delta x}^{\delta x} (x - x^*)\ \mathrm{arctg}\left[\frac{-d_{y1}}{x - x^*}\right] dx \tag{136}$$

$$I_{iv} = 0.5\left[\ d_{x2}^2\,\mathrm{arctg}\left[\frac{-d_{y1}}{-d_{x2}}\right] + d_{x2} d_{y1} + d_{y1}^2\,\mathrm{arctg}\left[\frac{-d_{y1}}{-d_{x2}}\right]\right. \tag{137}$$

$$- d_{x1}^2 \operatorname{arctg}\left[\frac{-d_{y1}}{-d_{x1}}\right] - d_{x1} d_{y1} - d_{y1}^2 \operatorname{arctg}\left[\frac{-d_{y1}}{-d_{x1}}\right]\Bigg]$$

d'où

$$2(I_{iii} - I_{iv}) = d_{x2}^2 \theta_2 - d_{y2}^2 \theta_3 + d_{x1}^2 \theta_4 - d_{y1}^2 \theta_1 + 4\delta x \delta y \qquad (138)$$

$$I_v = \int_{-\delta x}^{\delta x} dx = 2\delta x \qquad (139)$$

Après quelques manipulations, on obtient:

$$IV = \frac{1}{4\pi} [-d_{x2} d_{y1} \ln R_1^2 + d_{x1} d_{y1} \ln R_2^2 + d_{x2} d_{y2} \ln R_3^2 - d_{x1} d_{y2} \ln R_4^2 + d_{y1}^2 \theta_1 + d_{x2}^2 \theta_2 + d_{y2}^2 \theta_3 + d_{x1}^2 \theta_4 - 3\, \Delta x\, \Delta y] \qquad (140)$$

Cas 2: P est à l'intérieur du pavé (Fig. 18)

Pour effectuer cette intégrale, on doit utiliser les coordonnés polaires. De plus, on doit diviser le pavé en quatre triangles quelconques. Nous allons commencer par l'évaluation de l'intégrale sur le premier triangle.

Selon les paramètres apparaissant sur la fig. 18 et en utilisant la définition de G(r), l'intégrale sur le premier triangle devient:

$$IV_1 = \frac{1}{2\pi} \lim_{\epsilon \to 0} \int_{-\alpha_1}^{\alpha_2} \int_{\epsilon}^{d_{y1}/\cos\theta} \ln(r)\ r\ dr\ d\theta \qquad (141)$$

$$IV_1 = \frac{1}{8\pi} \int_{-\alpha_1}^{\alpha_2} \lim_{\epsilon \to 0} \left[2r^2 \ln(r) - r^2 \right]_{\epsilon}^{d_{y1}/\cos\theta} d\theta \tag{142}$$

$$IV_1 = \frac{1}{8\pi} \int_{-\alpha_1}^{\alpha_2} \frac{d_{y1}^2}{\cos^2\theta} \left[2\ln\left[\frac{d_{y1}}{\cos\theta}\right] - 1 \right] - 2\lim_{\epsilon \to 0}(\epsilon^2 \ln(\epsilon)) \quad d\theta \tag{143}$$

or la limite tend vers zéro, on obtient alors:

$$IV_1 = \frac{d_{y1}^2}{4\pi} \left[\int_{-\alpha_1}^{\alpha_2} \frac{\ln(d_{y1})-0.5}{\cos^2\theta} d\theta - \int_{-\alpha_1}^{\alpha_2} \frac{\ln(\cos\theta)}{\cos^2\theta} d\theta \right] \tag{144}$$

$$IV_1 = \frac{d_{y1}^2}{4\pi} \left[\ln(d_{y1})-0.5)\,tg\theta \Big]_{-\alpha_1}^{\alpha_2} - \int_{-\alpha_1}^{\alpha_2} \frac{\ln(\cos\theta)}{\cos^2\theta} d\theta \right] \tag{145}$$

En effectuant par partie, où $u = \ln[\cos\theta]$ (146)

$$dv = \sec^2\theta \; d\theta \tag{147}$$

$$du = -tg\theta \; d\theta \tag{148}$$

$$v = tg\theta \tag{149}$$

$$IV_1 = \frac{d_{y1}^2}{4\pi} \left[(\ln[d_{y1}]-0.5)\,tg\theta - tg\theta\,(\ln[\cos\theta]+1) + \theta \right]_{-\alpha_1}^{\alpha_2} \tag{150}$$

$$IV_1 = \frac{d_{y1}^2}{4\pi} \left[(\ln R_1 - 1.5)\,tg\alpha_2 + (\ln R_2 - 1.5)\,tg\alpha_1 + \alpha_1 + \alpha_2 \right] \tag{151}$$

or $\alpha_1 + \alpha_2 = \theta_1$ (152)

$$tg\alpha_1 = \frac{d_{x1}}{d_{y1}} \tag{153}$$

$$tg\alpha_2 = \frac{-d_{x2}}{d_{y1}} \tag{154}$$

$$IV_1 = \frac{1}{4\pi}\left[-(\ln R_1 - 1.5)\, d_{x2}\, d_{y1} + (\ln R_2 - 1.5)\, d_{x1}\, d_{y1} + d_{y1}^2\, \theta_1\right] \tag{155}$$

D'une façon similaire, on obtient :

$$IV_2 = \frac{1}{4\pi}\left[(\ln R_3 - 1.5)\, d_{x2}\, d_{y2} - (\ln R_1 - 1.5)\, d_{x2}\, d_{y1} + d_{x2}^2\, \theta_2\right] \tag{156}$$

$$IV_3 = \frac{1}{4\pi}\left[-(\ln R_4 - 1.5)\, d_{x1}\, d_{y2} + (\ln R_3 - 1.5)\, d_{x2}\, d_{y2} + d_{y2}^2\, \theta_3\right] \tag{157}$$

$$IV_4 = \frac{1}{4\pi}\left[(\ln R_2 - 1.5)\, d_{x1}\, d_{y1} - (\ln R_4 - 1.5)\, d_{x1}\, d_{y2} + d_{x1}^2\, \theta_4\right] \tag{158}$$

On additionne les quatres dernières relations pour obtenir:

$$IV = \frac{1}{4\pi}\,[-d_{x2}\, d_{y1} \ln R_1^2 + d_{x1}\, d_{y1} \ln R_2^2 + d_{x2}\, d_{y2} \ln R_3^2 - d_{x1}\, d_{y2} \ln R_4^2 + d_{y1}^2\, \theta_1 + d_{x2}^2\, \theta_2 + d_{y2}^2\, \theta_3 + d_{x1}^2\, \theta_4 - 3\, \Delta x\, \Delta y] \tag{159}$$

Cette équation est identique à la relation dérivée pour le premier cas.

ANNEXE 4

EQUIVALENCE DOUBLET-TOURBILLON

L'équivalence entre un doublet et un tourbillon est clairement expliquée par la relation suivante [10]:

$$\frac{1}{2\pi}\int_0^{l_j}\frac{\Phi\ y^*}{(x^*-\xi)^2+y^{*2}}d\xi = \frac{1}{2\pi}\Phi\theta\Big|_0^{l_j} - \frac{1}{2\pi}\int_0^{l_j}\Phi_\xi\theta\, d\xi \qquad (160)$$

Le terme de gauche de l'équation (160) est le potentiel induit par une distribution de doublet d'intensité Φ sur le panneau j.

Or, le potentiel induit par un tourbillon unitaire est donnée par:

$$\Phi = -\frac{1}{2\pi}\theta \qquad (161)$$

L'équation (160) montre donc qu'une distribution de doublet sur un panneau est équivalent à une distribution de tourbillon sur le panneau en question plus une paire de tourbillons ponctuels aux extrémités du panneau. L'intensité de la distribution de tourbillon est égale à la dérivée de l'intensité des doublets tandis que la puissance des tourbillons ponctuels est égale à la puissance des doublets aux extrémités.

ANNEXE 5

PROGRAMME

```
C*********************************************************************
C*********************************************************************
C*    PROGRAMME QUI PREDIT L'ECOULEMENT TRANSONIQUE AUTOUR D'UN      *
C*    PROFIL AERODYNAMIQUE EN UTILISANT UNE "FIELD PANEL METHOD".    *
C*                                                                   *
C*                                                                   *
C*    REF. : SINCLAIR, P.M.,"AN EXACT INTEGRAL(FIELD PANEL) METHOD   *
C*           FOR THE CALCULATION OF TWO-DIMENSIONAL TRANSONIC        *
C*           POTENTIAL FLOW AROUND COMPLEX CONFIGURATIONS",          *
C*           AERONAUTICAL JOURNAL, JUNE/JULY 1986.                   *
C*                                                                   *
C*    LA METHODE DES PANNEAUX QUE NOUS UTILISONS EST BASE SUR        *
C*    LA FORMULATION DE POTENTIEL DE VITESSE. CETTE METHODE UTILISE  *
C*    UNE DISTRIBUTION LINEAIRE  DE DOUBLET SUR CHAQUE PANNEAU.      *
C*                                                                   *
C*    REF. : MORAN, J.,"AN INTRODUCTION TO THEORETICAL AND           *
C*           COMPUTATIONAL AERODYNAMICS", JOHN WILEY & SONS,         *
C*           1984, PP. 269-273.                                      *
C*                                                                   *
C*********************************************************************
C*------------------------ PROGRAMME PRINCIPAL ----------------------*
C*********************************************************************
C
      PROGRAM FPANEL
C
C-----DECLARATION DES VARIABLES-------------------------------------
C
      PARAMETER(IPMAX=121,IMAX=120,JMAX=96,ITEMAX=31,NMAX=IMAX*JMAX)
      REAL*4 CPUSEC(2)
      REAL*4 ALPHA,MACH,CL,T,DELTA,BETA,PI,RADDEG,LX,LY,GAMMA,PRECD,
     &       DX,DY,DMAX,PRECPD
      REAL*4 WPREC(ITEMAX)
      REAL*4 VIT(IPMAX),X(IPMAX),Y(IPMAX),XC(IPMAX),PHI(IPMAX),
     &       STETA(IPMAX),CTETA(IPMAX),DS(IPMAX)
      REAL*4 K(IPMAX,IPMAX)
      REAL*4 XG(IMAX),YG(JMAX)
      REAL*4 UXP(IMAX,JMAX),UYP(IMAX,JMAX),SIGMA(IMAX,JMAX),
     &       V(IMAX,JMAX),U(IMAX,JMAX),PHIF(IMAX,JMAX),Q2(IMAX,JMAX),
     &       BF(IMAX,JMAX),EXT(IMAX,JMAX),UP(IMAX,JMAX),
     &       VP(IMAX,JMAX),MSQ(IMAX,JMAX),PHIP(IMAX,JMAX)
      REAL*4 AF(IMAX,JMAX,IPMAX)
      REAL*4 AP(IMAX,JMAX,IPMAX)
      INTEGER I,I1,I2,J1,J2,IGEOM,N,NX,NY,ITMAX,ITER,ITPMAX,ITERP,N1
      INTEGER ORDER(IPMAX)
      INTEGER IADR(IMAX,IMAX)
```

```
      CHARACTER*10 ARFOIL
      CHARACTER*64 FNAME
      LOGICAL DEBUG,GRAPH
C
C-----INITIALISATION DES VARIABLES-----------------------------------
C
      DATA PHIF/NMAX*0.0/,SIGMA/NMAX*0.0/,EXT/NMAX*1.0/,
     &     PHI/IPMAX*0.0/,UP/NMAX*1.0/,VP/NMAX*1.0/
C
C-----REGIONS COMMUNES----------------------------------------------
C
      COMMON/REG1/BF
      COMMON/REG2/AF
      COMMON/REG3/AP
C
      PI = 3.1415926535897932 38E0
      RADDEG = 180.E0/PI
C
C-----LECTURE DES DONNEES-------------------------------------------
C
      CALL INPUT(IGEOM,DEBUG,GRAPH,PRECD,ITMAX,PRECPD,ITPMAX,N,NX,NY,LX,
     &           LY,T,ALPHA,MACH,GAMMA,ARFOIL,X,Y)
C
      I2 = NX/LX
      I1 = NINT((NX-I2)/2.0) - 1
      I2 = I1 + I2 + 3
      J2 = NINT(T*NY/LY/2.0)+ 2
      J1 = NY/2 - J2
      J2 = NY/2 + J2 + 1
      ALPHA = ALPHA/RADDEG
      N1 = N + 1
      CALL INTERV(0)
C
C-----CALCUL DU MAILLAGE RECTANGULAIRE------------------------------
C
      CALL GRID(NX,NY,LX,LY,X(1),DX,DY,XG,YG)
C
C-----CALCUL DES CARACTERISTIQUES GEOMETRIQUES DU PROFIL------------
C
      CALL PTMIL(N,X,XC)
      CALL LONG(N,X,Y,DS)
      CALL COSIN(N,X,Y,CTETA,STETA,DELTA,BETA)
      CALL COEF(N,NX,NY,DX,DY,DS,X,Y,XG,YG,CTETA,STETA,DELTA,BETA,
     &          K,EXT,IADR)
C
C-----DECOMPOSITION LU DE LA MATRICE D'INFLUENCE K--------------------
C
      CALL LUDCMP(K,N1,ORDER)
      CALL INTERV(IC)
      CPUSEC(1) = FLOAT(IC)*1.0E-03
```

```
      ITER = 0
      WPREC(ITER) = 100
      CALL INTERV(0)
      GOTO 200
C
  100 CONTINUE
        ITER = ITER + 1
C
C-----ETAPE #3-----------------------------------------------------------
C
        CALL CALPHP(N,NX,NY,ALPHA,XG,YG,PHI,PHIP)
        CALL DIFCOR(NX,NY,DX,DY,EXT,PHIP,UP,VP)
        CALL DIFCOR(NX,NY,DX,DY,EXT,UP,UXP,UYP)
        IF(DEBUG)CALL IMPR(N,NX,NY,GAMMA,MACH,SIGMA,PHIF,U,V,UXP,UYP,
     &                     PHI)
C
C-----ETAPE #4, 5 ET 6---------------------------------------------------
C
        ITERP = 0
        DMAX = 0.0
        WPREC(ITER) = 0.0
  400   ITERP = ITERP + 1
          CALL SUPERP(NX,NY,MACH,GAMMA,UP,VP,U,V,Q2,MSQ)
          CALL SOURCE(ITER,NX,NY,DX,DY,U,V,UXP,UYP,Q2,MSQ,PHIF,SIGMA,
     &                WPREC(ITER),EXT)
          CALL CALPF(NX,NY,IADR,DMAX,SIGMA,PHIF)
          CALL CALPHF(NX,NY,I1,I2,J1,J2,EXT,PHIF)
          CALL DIFF(NX,NY,DX,DY,PHIF,U,V)
        IF((DMAX.GT.PRECPD).AND.(ITERP.LT.ITPMAX)) GOTO 400
        WRITE(6,*) ' ITER=',ITER,'   ITERP=',ITERP,'    DMAX=',DMAX
C
C-----ETAPES #7----------------------------------------------------------
C
        CALL FINF(N,NX,NY,SIGMA,PHI)
  200   DO 11 I = 1 , N
           PHI(I) =-PHI(I)-Y(I)*SIN(ALPHA)-X(I)*COS(ALPHA)
   11   CONTINUE
        PHI(N+1) = 0.E0
C
C-----ETAPE #2-----------------------------------------------------------
C
        CALL SOLVLU(K,PHI,N1,ORDER)
        IF(MACH.EQ.0.E0) GOTO 300
      IF ((WPREC(ITER).GT.PRECD).AND.(ITER.LT.ITMAX)) GOTO 100
C
      CALL INTERV(IC)
      CPUSEC(2) = FLOAT(IC)*1.0E-03
C
C-----CALCUL ET IMPRESSION DES CP-----------------------------------------
C
```

```
  300 CALL VITPOR(N,X(1),PHI,DS,VIT,CL)
      CALL RESULT(N,NX,NY,LX,LY,ARFOIL,ALPHA,MACH,GAMMA,PRECD,CL,XC,
     &            VIT,ITER,WPREC,CPUSEC)
C
       IF (DEBUG)CALL IMPR(N,NX,NY,GAMMA,MACH,SIGMA,PHIF,U,V,
     &                     UXP,UYP,PHI)
C
C------SORTIES GRAPHIQUES-----------------------------------------------
C
       IF (GRAPH) CALL WRESUL(N,NX,NY,LX,LY,ARFOIL,ALPHA,MACH,GAMMA,
     & PRECD,CL,X,Y,XC,XG,YG,VIT,U,V,UXP,UYP,SIGMA,ITER,WPREC,CPUSEC,
     & EXT,MSQ)
      STOP
      END
C
C*********************************************************************
C*    RECHERCHE DU PLUS GRAND PIVOT.                                 *
C*********************************************************************
C
      SUBROUTINE APVT(A,N,ORDER,JCOL)
C
C ------------------------------------------------------------------
C
C     SUBROUTINE APVT :
C                          THIS SUBROUTINE FINDS THE LARGEST ELEMENT FOR
C     PIVOT IN JCOL OF MATRIX A, PERFORMS INTERCHANGES OF ELEMENTS IN A
C     AND ALSO INTERCHANGES THE ELEMENTS IN THE ORDER VECTOR.
C
C ------------------------------------------------------------------
C
C     PARAMETERS ARE :
C
C     A      - MATRIX OF COEFFICIENTS WHOSE ROWS TO BE INTERCHANGED
C     N      - THE NUMBER OF EQUATIONS
C     ORDER  - INTEGER VECTOR TO HOLD ROW ORDERING
C     JCOL   - COLUMN OF A BEING SEARCHED FOR PIVOT ELEMENT
C
C ------------------------------------------------------------------
C
      PARAMETER (IPMAX=121)
      REAL*4 A(IPMAX,N)
      INTEGER N,JCOL,ORDER(N)
      REAL*4 SAVE,ANEXT,BIG
      INTEGER IPVT,JP1,IROW,KCOL,ISAVE
C
C ------------------------------------------------------------------
C
C  FIND PIVOT ROW, CONSIDERING ONLY THE ELEMENTS ON AND BELOW DIAGONAL
C
      IPVT = JCOL
```

```
      BIG = ABS(A(JCOL,JCOL))
      JP1 = JCOL + 1
      DO 10 IROW = JP1,N
        ANEXT = ABS(A(IROW,JCOL))
        IF ( ANEXT .GT. BIG ) IPVT = IROW
   10 CONTINUE
C
C ------------------------------------------------------------------
C
C  NOW INTERCHANGE ROW ELEMENTS IN THE ROW WHOSE NUMBER EQUALS JCOL WITH
C  THE PIVOT ROW UNLESS PIVOT ROW IS JCOL.
C
      IF ( IPVT .EQ. JCOL ) THEN
        RETURN
      END IF
      DO 20 KCOL = 1,N
        SAVE = A(JCOL,KCOL)
        A(JCOL,KCOL) = A(IPVT,KCOL)
        A(IPVT,KCOL) = SAVE
   20 CONTINUE
C
C ------------------------------------------------------------------
C
C  NOW SWITCH ELEMENTS IN THE ORDER VECTOR
C
      ISAVE = ORDER(JCOL)
      ORDER(JCOL) = ORDER(IPVT)
      ORDER(IPVT) = ISAVE
      RETURN
      END
C
C*********************************************************************
C*    CALCUL DU POTENTIEL INDUIT PAR LES SOURCE AU CENTRE DES PAVES. *
C*********************************************************************
C
      SUBROUTINE CALPF(NX,NY,IADR,DMAX,SIGMA,PHIF)
      PARAMETER (IMAX=120,JMAX=96)
      REAL*4 PHI,DMAX
      REAL*4 PHIF(IMAX,JMAX),BF(IMAX,JMAX),SIGMA(IMAX,JMAX)
      INTEGER NX,NY,K,L,I,J,II,JJ
      INTEGER IADR(IMAX,IMAX)
      COMMON/REG1/BF
      DO 999 L= 1,NX
        DO 999 K = 1,NY
          PHI = 0.E0
          DO 111 I = 1 , NX
            II = IADR(I,L)
            DO 111 J = 1 , NY
              JJ = IADR(J,K)
              PHI = PHI + SIGMA(I,J)*BF(II,JJ)
```

```
  111     CONTINUE
          DMAX = AMAX1(DMAX,ABS((PHI-PHIF(L,K))/(1.E0+PHIF(L,K))))
          PHIF(L,K) = PHI
  999 CONTINUE
      RETURN
      END
C
C*********************************************************************
C*    CORRECTION A L'INTERIEUR DU CORPS                              *
C*********************************************************************
C
      SUBROUTINE CALPHF(NX,NY,I1,I2,J1,J2,EXT,PHIF)
      PARAMETER (IPMAX=121,IMAX=120,JMAX=96)
      REAL*4 PHIF(IMAX,JMAX),EXT(IMAX,JMAX)
      INTEGER NX,NY,I,J,I1,I2,J1,J2
      DO 11 I = I1,I2
        DO 22 J = J1,NY/2
          IF (EXT(I,J).EQ.0.0) THEN
            PHIF(I,J) = 3.E0*(PHIF(I,J-1)-PHIF(I,J-2)) + PHIF(I,J-3)
          END IF
   22   CONTINUE
        DO 44 J = J2,NY/2+1,-1
          IF (EXT(I,J).EQ.0.0) THEN
            PHIF(I,J) = 3.E0*(PHIF(I,J+1)-PHIF(I,J+2)) + PHIF(I,J+3)
          END IF
   44   CONTINUE
   11 CONTINUE
      RETURN
      END
C
C*********************************************************************
C*    INFLUENCE DES DOUBLETS AU CENTRE DES PAVES                     *
C*********************************************************************
C
      SUBROUTINE CALPHP(N,NX,NY,ALPHA,XG,YG,PHI,PHIP)
      PARAMETER (IPMAX=121,IMAX=120,JMAX=96)
      REAL*4 ALPHA
      REAL*4 XG(IMAX),YG(JMAX)
      REAL*4 PHI(IPMAX)
      REAL*4 PHIP(IMAX,JMAX)
      REAL*4 AP(IMAX,JMAX,IPMAX)
      INTEGER N,NX,NY,I,J,L
      COMMON/REG3/AP
      DO 11 I = 1 , NX
        DO 11 J = 1 , NY
          PHIP(I,J) = YG(J)*SIN(ALPHA) + XG(I)*COS(ALPHA)
          DO 11 L = 1 , N+1
            PHIP(I,J) = PHIP(I,J) + AP(I,J,L)*PHI(L)
   11 CONTINUE
      RETURN
```

```
      END
C
C*********************************************************************
C*    GENERATION D'UN CERCLE DE RAYON T.                             *
C*********************************************************************
C
      SUBROUTINE CIRCLE(N,R,X,Y)
      PARAMETER (IPMAX=121)
      REAL*4 R,TETA,PI
      REAL*4 X(IPMAX),Y(IPMAX)
      INTEGER N,I
      PI = 3.1415926535897932 38E0
      TETA = 0.E0
      DO 11 I = 1 , N+1
        Y(I) = -R*SIN(TETA)
        X(I) =  R*(1.E0 + COS(TETA))
        TETA = TETA + 2.E0*PI/FLOAT(N)
   11 CONTINUE
      RETURN
      END
C
C*********************************************************************
C*    CALCUL DES COEFFICIENTS D'INFLUENCE.                           *
C*********************************************************************
C
      SUBROUTINE COEF(N,NX,NY,DX,DY,DS,X,Y,XG,YG,CTETA,STETA,
     &                DELTA,BETA,K,EXT,IADR)
      PARAMETER (IPMAX=121,IMAX=120,JMAX=96)
      REAL*4 TX,TY,DX,DY,DELTA,BETA,X1,Y1,X2,Y2
      REAL*8 R1,R2,E,PI,XI,YI,YMAX,YMIN,YU,YD,TETA1,
     &       TETA2,TETA3,TETA4,SQR1,SQR2,SQR3,SQR4,A1,A2,A3,A4,
     &       DDX,DDY,GG,ANG
      REAL*4 X(IPMAX),Y(IPMAX),DS(IPMAX),CTETA(IPMAX),STETA(IPMAX)
      REAL*4 K(IPMAX,IPMAX)
      REAL*4 XG(IMAX),YG(JMAX)
      REAL*4 BF(IMAX,JMAX),EXT(IMAX,JMAX)
      REAL*4 AF(IMAX,JMAX,IPMAX)
      REAL*4 AP(IMAX,JMAX,IPMAX)
      INTEGER I,J,L,N,M,NX,NY,IP,KK
      INTEGER IADR(IMAX,IMAX)
      COMMON/REG1/BF
      COMMON/REG2/AF
      COMMON/REG3/AP
      ANG(XI,YI,X1,Y1,X2,Y2) = DATAN2((YI-Y1)*(X2-XI)-(YI-Y2)*(X1-XI),
     &                                (YI-Y1)*(YI-Y2)+(X1-XI)*(X2-XI))
      PI = 3.1415926535897932 38E0
      DO 11 I = 1, N+1
        DO 11 J = 1 , N+1
          K(I,J) = 0.E0
   11 CONTINUE
```

```
      DO 77 I = 1,NX
        DO 77 J = 1,NY
          DO 77 L = 1,N+1
            AP(I,J,L) = 0.E0
77    CONTINUE
      DO 22 I = 1 , N
        DO 33 J = 1 , N
          IF(((I.NE.J).AND.(I.NE.J+1)).AND.((I.NE.1).OR.(J.NE.N)))THEN
            XI = (X(I)-X(J))*CTETA(J)+(Y(I)-Y(J))*STETA(J)
            YI =-(X(I)-X(J))*STETA(J)+(Y(I)-Y(J))*CTETA(J)
            R1 =SQRT((X(I)-X(J)  )**2+(Y(I)-Y(J)  )**2)
            R2 =SQRT((X(I)-X(J+1))**2+(Y(I)-Y(J+1))**2)
            DDY =((Y(I)-Y(J))*(X(J+1)-X(I))-(Y(I)-Y(J+1))*(X(J)-X(I)))
            DDX =((Y(I)-Y(J))*(Y(I)-Y(J+1))+(X(J)-X(I))*(X(J+1)-X(I)))
            E  = DATAN2(DDY,DDX)
            K(I,J)=K(I,J)+((1.D0-XI/DS(J))*E-YI/DS(J)*
     &                        DLOG(R2/R1))/(2.D0*PI)
            K(I,J+1)=(XI*E/DS(J)+YI/DS(J)*DLOG(R2/R1))/(2.D0*PI)
          END IF
33      CONTINUE
        IF (I.NE.1) THEN
          DDY=((Y(I)-Y(I-1))*(X(I+1)-X(I))-(Y(I)-Y(I+1))*(X(I-1)-X(I)))
          DDX=((Y(I)-Y(I-1))*(Y(I)-Y(I+1))+(X(I-1)-X(I))*(X(I+1)-X(I)))
          E  = DATAN2(DDY,DDX)
          K(I,I) = -1.D0+E/(2.D0*PI)
          DDY = Y(I)-Y(1)
          DDX = X(1)-X(I)
          E  = DATAN2(DDY,DDX)
          K(I,1) = K(I,1)-E/(2.D0*PI)
          K(I,N+1) = K(I,N+1)+E/(2.D0*PI)
        ELSE
          K(I,I) = -DELTA/(2.D0*PI)
          K(I,N+1) = -BETA/(2.D0*PI)
        END IF
22    CONTINUE
      K(N+1,1) = DS(N)
      K(N+1,2) = -DS(N)
      K(N+1,N+1) =-DS(1)
      K(N+1,N) = DS(1)
      DO 44 I = 1,NX
        DO 44 L = 1,NX
          IADR(I,L) = 1 + IABS(I-L)
44    CONTINUE
      IP = 0
      DO 55 I = 1,NX
        DO 55 J = 1,NY
          GG = 0.E0
          DO 66 L = 1 , N
            TX = (XG(I)-X(L))/(X(L+1)-X(L))
            TY = (YG(J)-Y(L))/(Y(L+1)-Y(L))
```

```
          IF ((TX.NE.TY).OR.((TX.NE.0.).AND.(TX.NE.1.))) THEN
            XI = (XG(I)-X(L))*CTETA(L)+(YG(J)-Y(L))*STETA(L)
            YI =-(XG(I)-X(L))*STETA(L)+(YG(J)-Y(L))*CTETA(L)
            R1 =SQRT((XG(I)-X(L))**2+(YG(J)-Y(L))**2)
            R2 =SQRT((XG(I)-X(L+1))**2+(YG(J)-Y(L+1))**2)
            DDY = ((YG(J)-Y(L))*(X(L+1)-XG(I))
     &            -(YG(J)-Y(L+1))*(X(L)-XG(I)))
            DDX = ((YG(J)-Y(L))*(YG(J)-Y(L+1))
     &            +(X(L)-XG(I))*(X(L+1)-XG(I)))
            E  = DATAN2(DDY,DDX)
            IF ((TX.EQ.TY).AND.((TX.LT.0.).OR.(TX.GT.1.))) THEN
              E = 0.E0
              YI= 0.E0
            END IF
            IF ((TX.EQ.TY).AND.(TX.GT.0.).AND.(TX.LT.1.)) THEN
              E = PI
              YI= 0.E0
            END IF
            GG = GG + E
            AP(I,J,L) = AP(I,J,L) + ((1.D0-XI/DS(L))*E-YI
     &                      /DS(L)*DLOG(R2/R1))/(2.D0*PI)
            AP(I,J,L+1)=(XI*E/DS(L)+YI/DS(L)*DLOG(R2/R1))
     &                      /(2.D0*PI)
          ELSE
            IF (TX.EQ.0.) IP = L
          END IF
66      CONTINUE
        IF (IP.NE.1) THEN
          DDY = YG(J)-Y(1)
          DDX = X(1)-XG(I)
          E  = DATAN2(DDY,DDX)
          AP(I,J,1) = AP(I,J,1)-E/(2.D0*PI)
          AP(I,J,N+1) = AP(I,J,N+1)+E/(2.D0*PI)
          IF (IP.NE.0) THEN
            DDY = ((YG(J)-Y(IP-1))*(X(IP+1)-XG(I))
     &            - (YG(J)-Y(IP+1))*(X(IP-1)-XG(I)))
            DDX = ((YG(J)-Y(IP-1))*(YG(J)-Y(IP+1))
     &            + (X(IP-1)-XG(I))*(X(IP+1)-XG(I)))
            E  = DATAN2(DDY,DDX)
            AP(I,J,IP) = E/(2.D0*PI)
          END IF
        ELSE
          AP(I,J,1) = -DELTA/(2.D0*PI)+ 1.D0
          AP(I,J,N+1) = -BETA/(2.D0*PI)
        END IF
        IP = 0
        IF(ABS(GG).GE.1.E-03)EXT(I,J)=0.E0
55 CONTINUE
   DO 111 L = 1 , N
     DO 111 I = 1 , NX
```

```
      DO 111 J = 1 , NY
        XI = X(L)-XG(I)
        YI = Y(L)-YG(J)
        A1 = DY/2.E0 + YI
        A2 = DX/2.E0 - XI
        A3 = DY/2.E0 - YI
        A4 = DX/2.E0 + XI
        SQR1 = A1*A1+A2*A2
        IF (SQR1.EQ.0.E0) SQR1=1.E0
        SQR2 = A1*A1+A4*A4
        IF (SQR2.EQ.0.E0) SQR2=1.E0
        SQR3 = A2*A2+A3*A3
        IF (SQR3.EQ.0.E0) SQR3=1.E0
        SQR4 = A3*A3+A4*A4
        IF (SQR4.EQ.0.E0) SQR4=1.E0
        TETA1 = ANG(XI,YI,-DX/2.E0,-DY/2.E0,DX/2.E0,-DY/2.E0)
        TETA2 = ANG(XI,YI,DX/2.E0,-DY/2.E0,DX/2.E0,DY/2.E0)
        TETA3 = ANG(XI,YI,DX/2.E0,DY/2.E0,-DX/2.E0,DY/2.E0)
        TETA4 = ANG(XI,YI,-DX/2.E0,DY/2.E0,-DX/2.E0,-DY/2.E0)
        AF(I,J,L) = (A1*A4*DLOG(SQR1)+A1*A2*DLOG(SQR2)+
     &                A2*A3*DLOG(SQR3)+A3*A4*DLOG(SQR4)+
     &                A1*A1*TETA1+A2*A2*TETA2+A3*A3*TETA3+
     &                A4*A4*TETA4-3.D0*DX*DY)/(4.D0*PI)
111 CONTINUE
    L = 1
    KK = 1
    DO 222 I = 1 , NX
      DO 222 J = 1 , NY
        XI = XG(L)-XG(I)
        YI = YG(KK)-YG(J)
        A1 = DY/2.E0 + YI
        A2 = DX/2.E0 - XI
        A3 = DY/2.E0 - YI
        A4 = DX/2.E0 + XI
        SQR1 = A1*A1+A4*A4
        IF (SQR1.EQ.0.E0) SQR1=1.E0
        SQR2 = A1*A1+A2*A2
        IF (SQR2.EQ.0.E0) SQR2=1.E0
        SQR3 = A2*A2+A3*A3
        IF (SQR3.EQ.0.E0) SQR3=1.E0
        SQR4 = A3*A3+A4*A4
        IF (SQR4.EQ.0.E0) SQR4=1.E0
        TETA1 = ANG(XI,YI,-DX/2.E0,-DY/2.E0,DX/2.E0,-DY/2.E0)
        TETA2 = ANG(XI,YI,DX/2.E0,-DY/2.E0,DX/2.E0,DY/2.E0)
        TETA3 = ANG(XI,YI,DX/2.E0,DY/2.E0,-DX/2.E0,DY/2.E0)
        TETA4 = ANG(XI,YI,-DX/2.E0,DY/2.E0,-DX/2.E0,-DY/2.E0)
        BF(I,J)=(A1*A4*DLOG(SQR1)+A1*A2*DLOG(SQR2)+A2*A3*DLOG(SQR3)+
     &            A3*A4*DLOG(SQR4)+A1*A1*TETA1+A2*A2*TETA2+A3*A3*TETA3+
     &            A4*A4*TETA4-3.D0*DX*DY)/(4.D0*PI)
222 CONTINUE
```

```
      RETURN
      END
C
C*********************************************************************
C*    CALCUL DE L'ANGLE D'INCLINAISON DE CHAQUE PANNEAU.             *
C*********************************************************************
C
      SUBROUTINE COSIN(N,X,Y,CTETA,STETA,DELTA,BETA)
      PARAMETER (IPMAX=121)
      REAL*4 DX,DY,DELTA,BETA
      REAL*4 X(IPMAX),Y(IPMAX),CTETA(IPMAX),STETA(IPMAX)
      INTEGER I,N
      PI = 3.14159265358979323 8E0
      DO 11 I = 1 , N
        DX = X(I+1) - X(I)
        DY = Y(I+1) - Y(I)
        STETA(I) = DY/SQRT(DX**2 + DY**2)
        CTETA(I) = DX/SQRT(DX**2 + DY**2)
   11 CONTINUE
      DELTA = PI - ATAN((Y(2)-Y(1))/(X(2)-X(1)))
      BETA  = PI + ATAN((Y(N+1)-Y(N))/(X(N+1)-X(N)))
      RETURN
      END
C
C
C*********************************************************************
C*    CALCUL DE UXP UYP UXXP ET UXYP                                 *
C*********************************************************************
C
      SUBROUTINE DIFCOR(NX,NY,DX,DY,E,PHI,U,V)
      PARAMETER (IMAX=120,JMAX=96)
      REAL*4 DX,DY,DCX,DCY,DMX,DPX,DMY,DPY,K1,K2,K3
      REAL*4 U(IMAX,JMAX),E(IMAX,JMAX),V(IMAX,JMAX)
      REAL*4 PHI(IMAX,JMAX)
      INTEGER I1,I2,J1,J2,I,J,NX,NY
      DO 22 I = 1,NX
        DO 22 J = 1,NY
          IF (E(I,J).NE.0) THEN
            K1 = E(I+1,J)*E(I-1,J)
            K2 = 1.E0-E(I-1,J)
            K3 = 1.E0-E(I+1,J)
            DCX = (PHI(I+1,J)-PHI(I-1,J))/2.E0/DX
            DMX = (3.E0*PHI(I,J)-4.E0*PHI(I-1,J)+PHI(I-2,J))/2.E0/DX
            DPX = (-3.E0*PHI(I,J)+4.E0*PHI(I+1,J)-PHI(I+2,J))/2.E0/DX
            U(I,J) = K1*DCX + K2*DPX+ K3*DMX
            K1 = E(I,J+1)*E(I,J-1)
            K2 = 1.E0-E(I,J-1)
            K3 = 1.E0-E(I,J+1)
            DCY = (PHI(I,J+1)-PHI(I,J-1))/2.E0/DY
            DMY = (3.E0*PHI(I,J)-4.E0*PHI(I,J-1)+PHI(I,J-2))/2.E0/DY
```

```
          DPY = (-3.E0*PHI(I,J)+4.E0*PHI(I,J+1)-PHI(I,J+2))/2.E0/DY
          V(I,J) = K1*DCY + K2*DPY+ K3*DMY
        END IF
   22 CONTINUE
      J1 = NY/2
      J2 =NY/2 + 1
      DO 33 I = 1,NX
        IF (E(I,J1).NE.0) THEN
          V(I,J1)=(3.E0*PHI(I,J1)-4.E0*PHI(I,J1-1)+PHI(I,J1-2))/2.E0/DY
        END IF
        IF (E(I,J2).NE.0) THEN
          V(I,J2)=(-3.E0*PHI(I,J2)+4.E0*PHI(I,J2+1)-PHI(I,J2+2))/2.E0/DY
        END IF
        V(I,1)=(-3.E0*PHI(I,1)+4.E0*PHI(I,2)-PHI(I,3))/2.E0/DY
        V(I,NY)=(3.E0*PHI(I,NY)-4.E0*PHI(I,NY-1)+PHI(I,NY-2))/2.E0/DY
   33 CONTINUE
      I1 = 1
      I2 = NX
      DO 44 J = 1,NY
        U(I1,J) = (-3.E0*PHI(I1,J)+4.E0*PHI(I1+1,J)-PHI(I1+2,J))/2.E0/DX
        U(I2,J) = (3.E0*PHI(I2,J)-4.E0*PHI(I2-1,J)+PHI(I2-2,J))/2.E0/DX
   44 CONTINUE
      RETURN
      END
C
C*********************************************************************
C*    CALCUL DE U, V, UX, UY                                         *
C*********************************************************************
C
      SUBROUTINE DIFF(NX,NY,DX,DY,PHI,U,V)
      PARAMETER (IMAX=120,JMAX=96)
      REAL*4 DX,DY
      REAL*4 PHI(IMAX,JMAX),U(IMAX,JMAX),V(IMAX,JMAX)
      INTEGER NX,NY,I,J
      DO 11 I = 2 , NX-1
        U(I,1)=(PHI(I+1,1)-PHI(I-1,1))/2.E0/DX
        U(I,NY)=(PHI(I+1,NY)-PHI(I-1,NY))/2.E0/DX
        V(I,1)=(-3.E0*PHI(I,1)+4.E0*PHI(I,2)-PHI(I,3))/2.E0/DY
        V(I,NY)=(3.E0*PHI(I,NY)-4.E0*PHI(I,NY-1)+PHI(I,NY-2))/2.E0/DY
         DO 11 J = 2 , NY-1
            U(I,J)=(PHI(I+1,J)-PHI(I-1,J))/2.E0/DX
            V(I,J)=(PHI(I,J+1)-PHI(I,J-1))/2.E0/DY
   11 CONTINUE
      DO 22 J = 2 , NY-1
        V(1,J)=(PHI(1,J+1)-PHI(1,J-1))/2.E0/DY
        V(NX,J)=(PHI(NX,J+1)-PHI(NX,J-1))/2.E0/DY
        U(1,J)=(-3.E0*PHI(1,J)+4.E0*PHI(2,J)-PHI(3,J))/2.E0/DX
        U(NX,J)=(3.E0*PHI(NX,J)-4.E0*PHI(NX-1,J)+PHI(NX-2,J))/2.E0/DX
   22 CONTINUE
      U(1,1)=(-3.E0*PHI(1,1)+4.E0*PHI(2,1)-PHI(3,1))/2.E0/DX
```

```
      V(1,1)=(-3.E0*PHI(1,1)+4.E0*PHI(1,2)-PHI(1,3))/2.E0/DY
      U(NX,1)=(3.E0*PHI(NX,1)-4.E0*PHI(NX-1,1)+PHI(NX-2,1))/2.E0/DX
      V(NX,1)=(-3.E0*PHI(NX,1)+4.E0*PHI(NX,2)-PHI(NX,3))/2.E0/DY
      U(1,NY)=(-3.E0*PHI(1,NY)+4.E0*PHI(2,NY)-PHI(3,NY))/2.E0/DX
      V(1,NY)=(3.E0*PHI(1,NY)-4.E0*PHI(1,NY-1)+PHI(1,NY-2))/2.E0/DY
      U(NX,NY)=(3.E0*PHI(NX,NY)-4.E0*PHI(NX-1,NY)+PHI(NX-2,NY))/2.E0/DX
      V(NX,NY)=(3.E0*PHI(NX,NY)-4.E0*PHI(NX,NY-1)+PHI(NX,NY-2))/2.E0/DY
      RETURN
      END
      END
C
C*********************************************************************
C*    CALCUL DE L'INFLUENCE DES SOURCES SUR LES PANNEAUX             *
C*********************************************************************
C
      SUBROUTINE FINF(N,NX,NY,SIGMA,PHI)
      PARAMETER(IPMAX=121,IMAX=120,JMAX=96)
      REAL*4 PHI(IPMAX)
      REAL*4 SIGMA(IMAX,JMAX)
      REAL*4 AF(IMAX,JMAX,IPMAX)
      INTEGER N,NX,NY,I,J,K,L
      COMMON/REG2/AF
      DO 11 L = N,1,-1
        PHI(L) = 0.0
        DO 11 I = 1 , NX
          DO 11 J = 1 , NY
            PHI(L) = PHI(L) + SIGMA(I,J)*AF(I,J,L)
   11 CONTINUE
      RETURN
      END
C
C*********************************************************************
C*    GENERATION DU MAILLAGE RECTANGULAIRE.                          *
C*********************************************************************
C
      SUBROUTINE GRID(NX,NY,LX,LY,CORD,DX,DY,XG,YG)
      PARAMETER (IMAX=120,JMAX=96)
      REAL*4 LX,LY,CORD,DX,DY,P1
      REAL*4 XG(IMAX),YG(JMAX)
      INTEGER NX,NY,I,J
      DY=LY*CORD/FLOAT(NY)
      DX=LX*CORD/FLOAT(NX)
      P1=((1.E0-LX)*CORD+DX)/2.E0
      DO 11 I = 1 , NX
         XG(I) = P1 + FLOAT(I-1)*DX
   11 CONTINUE
      P1=-(LY*CORD-DY)/2.E0
      DO 22 J = 1 , NY
         YG(J) = P1 + FLOAT(J-1)*DY
   22 CONTINUE
```

```
      RETURN
      END
C
C*********************************************************************
C*     IMPRESSION DE PHIF,U,V,UX,UY DANS LE CHAMPS                   *
C*********************************************************************
C
      SUBROUTINE IMPR(N,NX,NY,GAMMA,MACH,SIGMA,PHIF,U,V,UXP,UYP,PHI)
      PARAMETER (IPMAX=121,IMAX=120,JMAX=96)
      REAL*4 PHIF(IMAX,JMAX),U(IMAX,JMAX),V(IMAX,JMAX)
      REAL*4 UXP(IMAX,JMAX),UYP(IMAX,JMAX),SIGMA(IMAX,JMAX)
      REAL*4 PHI(IPMAX),GAMMA,MACH,A2,Q2
      INTEGER NX,NY
      WRITE(6,120)
      WRITE(6,130)
      IFST=1
      IB=IFST-10
    9 CONTINUE
      IB=IB+10
      IE=IB+9
      IE=MIN0(IE,N+1)
      WRITE(6,50)
      WRITE(6,20)(I,I=IB,IE)
      WRITE(6,50)
      J =  1
      WRITE(6,40)J,(PHI(I),I=IB,IE)
      IF(IE.LT.N+1) GO TO 9
      WRITE(6,50)
      WRITE(6,10)
      IFST=1
      JFST=1
      IB=IFST-10
    1 CONTINUE
      IB=IB+10
      IE=IB+9
      IE=MIN0(IE,NX)
      WRITE(6,50)
      WRITE(6,20)(I,I=IB,IE)
      WRITE(6,30)
      DO 11 J = NY , 1 , -1
         WRITE(6,40)J,(SIGMA(I,J),I=IB,IE)
   11 CONTINUE
      IF(IE.LT.NX) GO TO 1
      WRITE(6,50)
      WRITE(6,60)
      IFST=1
      JFST=1
      IB=IFST-10
    2 CONTINUE
      IB=IB+10
```

```
      IE=IB+9
      IE=MIN0(IE,NX)
      WRITE(6,50)
      WRITE(6,20)(I,I=IB,IE)
      WRITE(6,30)
      DO 22 J = NY , 1 , -1
         WRITE(6,40)J,(PHIF(I,J),I=IB,IE)
   22 CONTINUE
      IF(IE.LT.NX) GO TO 2
      WRITE(6,50)
      WRITE(6,70)
      IFST=1
      JFST=1
      IB=IFST-10
    3 CONTINUE
      IB=IB+10
      IE=IB+9
      IE=MIN0(IE,NX)
      WRITE(6,50)
      WRITE(6,20)(I,I=IB,IE)
      WRITE(6,30)
      DO 33 J = NY , 1 , -1
         WRITE(6,40)J,(U(I,J),I=IB,IE)
   33 CONTINUE
      IF(IE.LT.NX) GO TO 3
      WRITE(6,50)
      WRITE(6,80)
      IFST=1
      JFST=1
      IB=IFST-10
    4 CONTINUE
      IB=IB+10
      IE=IB+9
      IE=MIN0(IE,NX)
      WRITE(6,50)
      WRITE(6,20)(I,I=IB,IE)
      WRITE(6,30)
      DO 44 J = NY , 1 , -1
         WRITE(6,40)J,(V(I,J),I=IB,IE)
   44 CONTINUE
      IF(IE.LT.NX) GO TO 4
      WRITE(6,50)
      WRITE(6,90)
      IFST=1
      JFST=1
      IB=IFST-10
    5 CONTINUE
      IB=IB+10
      IE=IB+9
      IE=MIN0(IE,NX)
```

```
      WRITE(6,50)
      WRITE(6,20)(I,I=IB,IE)
      WRITE(6,30)
      DO 55 J = NY , 1 , -1
         WRITE(6,40)J,(UXP(I,J),I=IB,IE)
   55 CONTINUE
      IF(IE.LT.NX) GO TO 5
      WRITE(6,50)
      WRITE(6,100)
      IFST=1
      JFST=1
      IB=IFST-10
    6 CONTINUE
      IB=IB+10
      IE=IB+9
      IE=MIN0(IE,NX)
      WRITE(6,50)
      WRITE(6,20)(I,I=IB,IE)
      WRITE(6,30)
      DO 66 J = NY , 1 , -1
         WRITE(6,40)J,(UYP(I,J),I=IB,IE)
   66 CONTINUE
      IF(IE.LT.NX) GO TO 6
      WRITE(6,50)
      WRITE(6,150)
      IFST=1
      JFST=1
      IB=IFST-10
   12 CONTINUE
      IB=IB+10
      IE=IB+9
      IE=MIN0(IE,NX)
      WRITE(6,50)
      WRITE(6,20)(I,I=IB,IE)
      WRITE(6,30)
      DO 88 J = NY , 1 , -1
        WRITE(6,40)J,(SQRT((U(I,J)*U(I,J) + V(I,J)*V(I,J))/
     &                (1.D0/MACH/MACH + (GAMMA-1.D0)*(1.D0-
     &                (U(I,J)*U(I,J) + V(I,J)*V(I,J)))/2.D0)),I=IB,IE)
   88 CONTINUE
      IF(IE.LT.NX) GO TO 12
   10 FORMAT(1X,'SIGMA')
   20 FORMAT(1X,4H I =,I6,9I9)
   30 FORMAT(1X,1HJ)
   40 FORMAT(1X,I2,3X,10(1PE9.2))
   50 FORMAT(1X,1H )
   60 FORMAT(1X,'PHIF')
   70 FORMAT(1X,'U')
   80 FORMAT(1X,'V')
   90 FORMAT(1X,'UX')
```

```
  100 FORMAT(1X,'UY')
  120 FORMAT(1H1)
  130 FORMAT(1X,'PHI')
  150 FORMAT(1X,'Mach')
      RETURN
      END
C
C*********************************************************************
C*    LECTURE DES DONNEES ET GENERATION DE LA GEOMETRIE.             *
C*********************************************************************
C
      SUBROUTINE INPUT(IGEOM,DEBUG,GRAPH,PRECD,ITMAX,PRECPD,ITPMAX,N,NX,
     &              NY,LX,LY,T,ALPHA,MACH,GAMMA,ARFOIL,X,Y)
      PARAMETER (IPMAX=121)
      REAL*4 ALPHA,MACH,T,LX,LY,GAMMA,PRECD,TETA,PI,TETAI,XM,PRECPD
      REAL*4 X(IPMAX),Y(IPMAX)
      INTEGER I,IGEOM,N,N1,N2,NX,NY,ITMAX,ITPMAX
      CHARACTER*10 ARFOIL
      CHARACTER*64 VNAME
      LOGICAL DEBUG,GRAPH
      PI = 3.14159265358979323 8E0
      READ(5,20) VNAME
      READ(5,10) ARFOIL
      READ(5,20) VNAME
      READ(5,*) IGEOM,DEBUG,GRAPH
      READ(5,20) VNAME
      READ(5,*) PRECD,ITMAX
      READ(5,20) VNAME
      READ(5,*) PRECPD,ITPMAX
      READ(5,20) VNAME
      READ(5,*) T
      READ(5,20) VNAME
      READ(5,*) N1,N2,NX,NY,TETAI,XM
      N = N1+N2
      TETAI=TETAI*PI/180.E0
      READ(5,20) VNAME
      READ(5,*) LX,LY
      READ(5,20) VNAME
      READ(5,*) ALPHA,MACH,GAMMA
      IF (IGEOM.EQ.1) THEN
        X(1) = 1.0
        IF (XM.EQ.0.0) THEN
          DO 11 I = 2 , N/2
            TETA = 2.E0*(PI-TETAI)*FLOAT(I-1)/FLOAT(N) + TETAI
            X(I) = (1.E0 + COS(TETA))/(1.E0+COS(TETAI))
   11     CONTINUE
        ELSE
          DO 22 I = 2 , N1/2
            TETA = 2.E0*(PI-TETAI)*FLOAT(I-1)/FLOAT(N1) + TETAI
            X(I) = (1.E0 + COS(TETA))*(1.E0-XM)/(1.E0+COS(TETAI))+ XM
```

```
   22      CONTINUE
           X(N1/2+1) = XM
           DO 33 I = 2 , N2/2
             TETA = 2.E0*(PI-TETAI)*FLOAT(I-1)/FLOAT(N2) + TETAI
             X(I+N1/2) = (1.E0 + COS(TETA))*XM/(1.E0+COS(TETAI))
   33      CONTINUE
         END IF
         X(N/2+1) = 0.E0
         CALL NACA(N,T,X,Y)
      END IF
      IF (IGEOM.EQ.2) CALL CIRCLE(N,T,X,Y)
   10 FORMAT(A)
   20 FORMAT(A)
      RETURN
      END
C
C*********************************************************************
C*    CALCUL DE LA LONGUEUR DE CHAQUE PANNEAU.                       *
C*********************************************************************
C
      SUBROUTINE LONG(N,X,Y,DS)
      PARAMETER (IPMAX=121)
      REAL*4 X(IPMAX),Y(IPMAX),DS(IPMAX)
      INTEGER I,N
      DO 11 I = 1 , N
         DS(I)=SQRT(((X(I+1)-X(I)))**2 + ((Y(I+1)-Y(I)))**2)
   11 CONTINUE
      RETURN
      END
C
C*********************************************************************
C*    CALCUL DES MATRICES L & U.                                     *
C*********************************************************************
C
      SUBROUTINE LUDCMP(A,N,ORDER)
C
C ------------------------------------------------------------------
C
C     SUBROUTINE LUDCMP :
C                          THIS SUBROUTINE COMPUTES THE L AND U TRIANGULAR
C     MATRICES EQUIVALENT TO THE A MATRIX, SUCH THAT LU = A. THESE
C     MATRICES ARE RETURNED IN THE SPACE OF A, IN COMPACT FORM. THE U
C     MATRIX HAS ONES ON ITS DIAGONAL. PARTIAL PIVOTING IS USED TO GIVE
C     MAXIMUM VALUED ELEMENTS ON THE DIAGONAL OF L.THE ORDER OF THE ROWS
C     AFTER PIVOTING IS RETURNED IN VECTORS IN THE INTEGER VECTOR ORDER.
C     THIS SHOULD BE USED TO REORDER R.H.S. VECTORS BEFORE SOLVING THE
C     SYSTEM AX = B.
C
C ------------------------------------------------------------------
C
```

```
C     PARAMETERS ARE :
C
C     A      - THE N X N MATRIX OF COEFFICIENTS
C     N      - THE NUMBER OF EQUATIONS
C     ORDER - INTEGER VECTOR HOLDING ROW ORDER AFTER PIVOTING
C
C     THIS ROUTINE CALLS A SUBROUTINE APVT TO LOCATE THE PIVOT ROW AND
C     MAKE INTERCHANGES.
C ---------------------------------------------------------------------
C
      PARAMETER (IPMAX=121)
      REAL*4 A(IPMAX,N)
      INTEGER N,ORDER(N)
      REAL*4 SUM
      INTEGER I,KCOL,NM1,JM1,JCOL,JP1,IROW
C
C ---------------------------------------------------------------------
C
C  ESTABLISH INITIAL ORDERING IN ORDER VECTOR
C
      DO 10 I = 1,N
        ORDER(I) = I
   10 CONTINUE
C
C  DO PIVOTING FOR FIRST COLUMN BY CALL TO SUBROUTINE APVT
C
      CALL APVT(A,N,ORDER,1)
C
C ---------------------------------------------------------------------
C
C  IF PIVOT ELEMENT VERY SMALL, PRINT ERROR MESSAGE AND RETURN
C
      IF ( ABS(A(1,1)) .LT. 1.0D-5 ) THEN
        WRITE(6,100)
        RETURN
      END IF
C
C  NOW COMPUTE ELEMENTS FOR FIRST ROW OF U
C
      DO 20 KCOL = 2,N
        A(1,KCOL) = A(1,KCOL) / A(1,1)
   20 CONTINUE
C
C ---------------------------------------------------------------------
C
C  COMPLETE THE COMPUTING OF L AND U ELEMENTS. THE GENERAL PLAN IS TO
C  COMPUTE A COLUMN OF L'S, THEN CALL APVT TO INTERCHANE ROWS, AND THEN
C  GET A ROW OF U'S.
C
      NM1 = N - 1
```

```
      DO 80 JCOL = 2,NM1
C
C  FIRST COMPUTE A COLUMN OF L'S
C
        JM1 = JCOL - 1
        DO 50 IROW = JCOL,N
          SUM = 0
          DO 40 KCOL = 1,JM1
            SUM = SUM + A(IROW,KCOL)*A(KCOL,JCOL)
   40     CONTINUE
          A(IROW,JCOL) = A(IROW,JCOL) - SUM
   50   CONTINUE
C
C ---------------------------------------------------------------------
C
C  NOW INTERCHANGE ROWS IF NEED TO BE, THEN TEST FOR TOO SMALL PIVOT
C
        CALL APVT(A,N,ORDER,JCOL)
        IF ( ABS(A(JCOL,JCOL)) .LT. 1.0D-5 ) THEN
          WRITE(6,100)
          RETURN
        END IF
C
C  NOW WE GET A ROW OF U'S
C
        JP1 = JCOL + 1
        DO 70 KCOL = JP1,N
          SUM = 0
          DO 60 IROW = 1,JM1
            SUM = SUM + A(JCOL,IROW)*A(IROW,KCOL)
   60     CONTINUE
        A(JCOL,KCOL) = ( A(JCOL,KCOL) - SUM ) / A(JCOL,JCOL)
   70   CONTINUE
   80 CONTINUE
C
C ---------------------------------------------------------------------
C
C  STILL NEED TO GET LAST ELEMENT IN L MATRIX
C
      SUM = 0
      DO 90 KCOL = 1,NM1
        SUM = SUM + A(N,KCOL)*A(KCOL,N)
   90 CONTINUE
      A(N,N) = A(N,N) - SUM
      RETURN
C
  100 FORMAT(/,' VERY SMALL PIVOT ELEMENT INDICATES A NEARLY',
     &       'SINGULAR MATRIX. ')
      END
C
```

```
C*********************************************************************
C*    GENERATION DU PROFIL NACA00XX                                  *
C*********************************************************************
C
      SUBROUTINE NACA(N,T,X,Y)
      PARAMETER (IPMAX=121)
      REAL*4 T,X(IPMAX),Y(IPMAX)
      INTEGER I,N
      DO 11 I = 1 , N/2
         Y(I) =-(0.2969E0*SQRT(X(I))-0.1260E0*X(I)-0.3537E0*X(I)**2+
     &           0.2843E0*X(I)**3-0.1015E0*X(I)**4)*T/0.2E0
         X(N+2-I) = X(I)
         Y(N+2-I) =-Y(I)
   11 CONTINUE
      Y(N/2+1) = 0.E0
      Y(1) = 0.E0
      Y(N+1) = 0.E0
      RETURN
      END
C
C*********************************************************************
C*    CALCUL DES POINTS MILIEUX                                      *
C*********************************************************************
C
      SUBROUTINE PTMIL(N,X,XC)
      PARAMETER (IPMAX=121)
      REAL*4 X(IPMAX),XC(IPMAX)
      INTEGER I,N
      DO 11 I = 1 , N
         XC(I) = (X(I+1) + X(I))/2.E0
   11 CONTINUE
      RETURN
      END
C
C*********************************************************************
C*    IMPRESSION DES RESULTATS.                                      *
C*********************************************************************
C
      SUBROUTINE RESULT(N,NX,NY,LX,LY,ARFOIL,ALPHA,MACH,GAMMA,PRECD,
     &                  CL,XC,VIT,ITER,WPREC,CPUSEC)
      PARAMETER (IPMAX=121,ITEMAX=31)
      REAL*4 CPUSEC(2)
      REAL*4 XC(IPMAX),VIT(IPMAX),WPREC(ITEMAX)
      REAL*4 PI,ALPHA,MACH,CL,RADDEG,GAMMA,CPDO,CPUP,K1,K2,K3,PRECD,
     &       LX,LY
      INTEGER I,N,ITER
      CHARACTER*10 ARFOIL
      PI = 3.1415926535897932 38E0
      RADDEG = 180.E0/PI
      IF (MACH.NE.0.E0) THEN
```

```
      K1 = 2.E0/GAMMA/MACH/MACH
      K2 = (GAMMA-1.E0)*MACH*MACH/2.E0
      K3 = GAMMA/(GAMMA-1.E0)
    END IF
    IF (MACH.NE.0.E0) THEN
      WRITE(6,20)
      WRITE(6,30) ARFOIL
      WRITE(6,40) ALPHA*RADDEG,MACH,GAMMA
      WRITE(6,50) N
      WRITE(6,60) NX,NY,LX,LY
      WRITE(6,110) PRECD
      WRITE(6,120) (CPUSEC(I),I=1,2)
      WRITE(6,140)
      DO 11 I = 1,ITER
         WRITE(6,130) I,WPREC(I)
11    CONTINUE
    ELSE
      WRITE(6,25)
      WRITE(6,30) ARFOIL
      WRITE(6,45) ALPHA*RADDEG
      WRITE(6,50) N
    END IF
    WRITE(6,10)
    WRITE(6,70)
    WRITE(6,80)
    DO 22 I = 1 , N/2
       IF (MACH.NE.0.E0) THEN
         CPDO = K1*((1.E0+K2*(1.E0-VIT(I)**2))**K3-1.E0)
         CPUP = K1*((1.E0+K2*(1.E0-VIT(N+1-I)**2))**K3-1.E0)
       ELSE
         CPDO = 1.E0-VIT(I)*VIT(I)
         CPUP = 1.E0-VIT(N+1-I)*VIT(N+1-I)
       END IF
       WRITE(6,90) XC(I),CPDO,CPUP
22  CONTINUE
    WRITE(6,100) CL
10  FORMAT(1H1)
20  FORMAT(1X,'PROGRAMME FIELD PANEL (FORMULATION PAR POTEN. DE VIT.',
  &        ' LINEAIRE)',/)
25  FORMAT(1X,'PROGRAMME PANEL (FORMULATION PAR POTEN. DE VIT.',
  &        ' LINEAIRE)',/)
30  FORMAT(1X,'PROFIL ',A10)
40  FORMAT(1X,'    ANGLE D''ATTAQUE            : ',F6.3,/,1X,
  &           '    NOMBRE DE MACH             : ',F6.3,/,1X,
  &           '    GAMMA                      : ',F6.3)
45  FORMAT(1X,'    ANGLE D''ATTAQUE            : ',F6.3,/)
50  FORMAT(1X,'    NOMBRE DE PANNEAUX         : ',I3,/)
60  FORMAT(1X,'DOMAINE',/,1X,'    NOMBRE DE DIVISIONS EN X : ',I3,/,
  &        1X,'    NOMBRE DE DIVISIONS EN Y : ',I3,/,
  &        1X,'    LONGUEUR EN X / CORDE    : ',F4.1,/,
```

```
     &        1X,'   LONGUEUR EN Y / CORDE     : ',F4.1,/)
   70 FORMAT('    X          CP          CP')
   80 FORMAT('               DOWN        UP',/)
   90 FORMAT(1X,F6.4,2F11.4)
  100 FORMAT(//,1X,'CL : ',F6.3)
  110 FORMAT(1X,'PRECISION',19X,': ',E10.3,//)
  120 FORMAT(1X,'TEMPS (SEC) POUR EVALUER LES COEF.=',F7.2,
     &        /,1X,'TEMPS (SEC) POUR LES ITERATIONS   =',F7.2)
  130 FORMAT(1X,'ITERATION # ',I2,
     &           '      PREC = ',E10.3)
  140 FORMAT(/,1X,'TRACE DE LA CONVERGENCE',/)
      RETURN
      END
C
C*********************************************************************
C*    CALCUL DE LA SOLUTION DE (LU)X = B.                            *
C*********************************************************************
C
      SUBROUTINE SOLVLU(LU,B,N,ORDER)
C
C ------------------------------------------------------------------
C
C     SUBROUTINE SOLVLU :
C                          THIS SUBROUTINE IS USED TO FIND THE SOLUTION
C    TO A SYSTEM OF EQUATIONS, AX = B, AFTER THE LU EQUIVALENT OF A HAS
C    BEEN FOUND. BEFORE USING THIS ROUTINE, THE VECTOR B SHOULD BE
C    SCALED IF MATRIX A WAS SCALED, USING THE SAME SCALE FACTORS. WITHIN
C    THIS ROUTINE, THE ELEMENTS OF B ARE REARRANGED IN THE SAME WAY THAT
C    THE ROWS OF A WERE INTERCHANGED, USING THE ORDER VECTOR WHICH HOLDS
C    THE ROW ORDERINGS. THE SOLUTION IS RETURNED IN B.
C
C ------------------------------------------------------------------
C
C     PARAMETERS ARE :
C
C     LU    - THE LU EQUIVALENT OF THE COEFFICIENT MATRIX
C     B     - THE VECTOR OF RIGHT HAND SIDES
C     N     - NUMBER OF EQUATIONS
C     ORDER -  INTEGER ARRAY OF ROW ORDER AS ARRANGED DURING PIVOTING
C
C ------------------------------------------------------------------
C
      PARAMETER (IPMAX=121)
      REAL*4 LU(IPMAX,N),B(N),X(IPMAX)
      INTEGER N,ORDER(N)
      REAL*4 SUM
      INTEGER I,J,JCOL,IROW,NP1,NVBL,IM1
C
C ------------------------------------------------------------------
C
```

```
C  REARRANGE THE ELEMENTS OF THE B VECTOR. X IS USED TO HOLD THEM.
C
      DO 10 I = 1,N
        J = ORDER(I)
        X(I) = B(J)
   10 CONTINUE
C
C ---------------------------------------------------------------------
C
C  COMPUTE THE B' VECTOR, STORING BACK IN X
C
      X(1) = X(1) / LU(1,1)
      DO 50 IROW = 2,N
        IM1 = IROW - 1
        SUM = 0
        DO 40 JCOL = 1,IM1
          SUM = SUM + LU(IROW,JCOL)*X(JCOL)
   40   CONTINUE
        X(IROW) = (X(IROW) - SUM) / LU(IROW,IROW)
   50 CONTINUE
C
C ---------------------------------------------------------------------
C
C  NOW GET THE SOLUTION VECTOR. X(N) = X(N) ALREADY
      DO 70 IROW = 2,N
        NVBL = N - IROW + 1
        SUM = 0
        NP1 = NVBL + 1
        DO 60 JCOL = NP1,N
          SUM = SUM + LU(NVBL,JCOL)*X(JCOL)
   60   CONTINUE
        X(NVBL) = X(NVBL) - SUM
   70 CONTINUE
      DO 80 IROW = 1,N
        B(ORDER(IROW)) = X(IROW)
   80 CONTINUE
      RETURN
      END
C
C*********************************************************************
C*    CALCUL DE LA PUISSANCE DES SOURCES DANS LE CHAMPS.             *
C*********************************************************************
C
      SUBROUTINE SOURCE(ITER,NX,NY,DX,DY,U,V,UXP,UYP,Q2,MSQ,PHIF,SIGMA,
     &                  WPREC,EXT)
      PARAMETER (IMAX=120,JMAX=96)
      REAL*4 S,WPREC,MU,BE,GA,CX,CY,CXY,A,B,C,D,E,F,G,H,II,JJ,K,L,N,
     &       DX,DY,PHISSP,BETA2,MU1,BE1,GA1,PHISSF,PXX,PXY,MACH,GAMMA
      REAL*4 U(IMAX,JMAX),V(IMAX,JMAX),UXP(IMAX,JMAX),
     &       UYP(IMAX,JMAX),SIGMA(IMAX,JMAX),MSQ(IMAX,JMAX),
```

```
  &        PHIF(IMAX,JMAX),EXT(IMAX,JMAX),Q2(IMAX,JMAX)
   INTEGER NX,NY,I,ITER,J
   DO 11 I = 2,NX-1
      J = 1
      PHISSP = (U(I,J)*U(I,J)-V(I,J)*V(I,J))*UXP(I,J) + 2.E0*
  &             U(I,J)*V(I,J)*UYP(I,J)
      PXX = (PHIF(I+1,J)-2.0*PHIF(I,J)+PHIF(I-1,J))/DX/DX
      PXY = (PHIF(I+1,J+1)-PHIF(I+1,J)-PHIF(I-1,J+1)+PHIF(I-1,J))/
  &         2.0/DX/DY
      S  = ((U(I,J)*U(I,J)-V(I,J)*V(I,J))*PXX+2.0*U(I,J)*V(I,J)*PXY+
  &         PHISSP)/(Q2(I,J)/MSQ(I,J)-V(I,J)*V(I,J))
      WPREC = AMAX1(ABS((SIGMA(I,J)-S)/(1.E0+S)),WPREC)
      SIGMA(I,J) = S
      J = NY
      PHISSP = (U(I,J)*U(I,J)-V(I,J)*V(I,J))*UXP(I,J) + 2.E0*
  &             U(I,J)*V(I,J)*UYP(I,J)
      PXX = (PHIF(I+1,J)-2.0*PHIF(I,J)+PHIF(I-1,J))/DX/DX
      PXY = (PHIF(I+1,J)-PHIF(I+1,J-1)-PHIF(I-1,J)+PHIF(I-1,J-1))/
  &         2.0/DX/DY
      S  = ((U(I,J)*U(I,J)-V(I,J)*V(I,J))*PXX+2.0*U(I,J)*V(I,J)*PXY+
  &         PHISSP)/(Q2(I,J)/MSQ(I,J)-V(I,J)*V(I,J))
      WPREC = AMAX1(ABS((SIGMA(I,J)-S)/(1.E0+S)),WPREC)
      SIGMA(I,J) = S
11 CONTINUE
   DO 22 J = 2,NY-1
      I = 1
      PHISSP = (U(I,J)*U(I,J)-V(I,J)*V(I,J))*UXP(I,J) + 2.E0*
  &             U(I,J)*V(I,J)*UYP(I,J)
      PXX = (PHIF(I,J+1)-2.0*PHIF(I,J)+PHIF(I,J-1))/DY/DY
      PXY = (PHIF(I+1,J+1)-PHIF(I+1,J-1)-PHIF(I,J+1)+PHIF(I,J-1))/
  &         2.0/DX/DY
      S  = ((V(I,J)*V(I,J)-U(I,J)*U(I,J))*PXX+2.0*U(I,J)*V(I,J)*PXY+
  &         PHISSP)/(Q2(I,J)/MSQ(I,J)-U(I,J)*U(I,J))
      WPREC = AMAX1(ABS((SIGMA(I,J)-S)/(1.E0+S)),WPREC)
      SIGMA(I,J) = S
      I = NX
      PHISSP = (U(I,J)*U(I,J)-V(I,J)*V(I,J))*UXP(I,J) + 2.E0*
  &             U(I,J)*V(I,J)*UYP(I,J)
      PXX = (PHIF(I,J+1)-2.0*PHIF(I,J)+PHIF(I,J-1))/DY/DY
      PXY = (PHIF(I,J+1)-PHIF(I,J-1)-PHIF(I-1,J+1)+PHIF(I-1,J-1))/
  &         2.0/DX/DY
      S  = ((V(I,J)*V(I,J)-U(I,J)*U(I,J))*PXX+2.0*U(I,J)*V(I,J)*PXY+
  &         PHISSP)/(Q2(I,J)/MSQ(I,J)-U(I,J)*U(I,J))
      WPREC = AMAX1(ABS((SIGMA(I,J)-S)/(1.E0+S)),WPREC)
      SIGMA(I,J) = S
22 CONTINUE
   I = 1
   J = 1
   PHISSP = (U(I,J)*U(I,J)-V(I,J)*V(I,J))*UXP(I,J) + 2.E0*
  &             U(I,J)*V(I,J)*UYP(I,J)
```

```
 PXX = (PHIF(I,J)-2.0*PHIF(I+1,J)+PHIF(I+2,J))/DX/DX
 PXY = (PHIF(I+1,J+1)-PHIF(I+1,J)-PHIF(I,J+1)+PHIF(I,J))/DX/DY
 S  = ((U(I,J)*U(I,J)-V(I,J)*V(I,J))*PXX+2.0*U(I,J)*V(I,J)*PXY+
&        PHISSP)/(Q2(I,J)/MSQ(I,J)-V(I,J)*V(I,J))
 WPREC = AMAX1(ABS((SIGMA(I,J)-S)/(1.E0+S)),WPREC)
 SIGMA(I,J) = S
 I = 1
 J = NY
 PHISSP = (U(I,J)*U(I,J)-V(I,J)*V(I,J))*UXP(I,J) + 2.E0*
&            U(I,J)*V(I,J)*UYP(I,J)
 PXX = (PHIF(I,J)-2.0*PHIF(I+1,J)+PHIF(I+2,J))/DX/DX
 PXY = (PHIF(I+1,J)-PHIF(I+1,J-1)-PHIF(I,J)+PHIF(I,J-1))/DX/DY
 S  = ((U(I,J)*U(I,J)-V(I,J)*V(I,J))*PXX+2.0*U(I,J)*V(I,J)*PXY+
&        PHISSP)/(Q2(I,J)/MSQ(I,J)-V(I,J)*V(I,J))
 WPREC = AMAX1(ABS((SIGMA(I,J)-S)/(1.E0+S)),WPREC)
 SIGMA(I,J) = S
 I = NX
 J = 1
 PHISSP = (U(I,J)*U(I,J)-V(I,J)*V(I,J))*UXP(I,J) + 2.E0*
&            U(I,J)*V(I,J)*UYP(I,J)
 PXX = (PHIF(I,J)-2.0*PHIF(I-1,J)+PHIF(I-2,J))/DX/DX
 PXY = (PHIF(I,J+1)-PHIF(I,J)-PHIF(I-1,J+1)+PHIF(I-1,J))/DX/DY
 S  = ((U(I,J)*U(I,J)-V(I,J)*V(I,J))*PXX+2.0*U(I,J)*V(I,J)*PXY+
&        PHISSP)/(Q2(I,J)/MSQ(I,J)-V(I,J)*V(I,J))
 WPREC = AMAX1(ABS((SIGMA(I,J)-S)/(1.E0+S)),WPREC)
 SIGMA(I,J) = S
 I = NX
 J = NY
 PHISSP = (U(I,J)*U(I,J)-V(I,J)*V(I,J))*UXP(I,J) + 2.E0*
&            U(I,J)*V(I,J)*UYP(I,J)
 PXX = (PHIF(I,J)-2.0*PHIF(I-1,J)+PHIF(I-2,J))/DX/DX
 PXY = (PHIF(I,J)-PHIF(I,J-1)-PHIF(I-1,J)+PHIF(I-1,J-1))/DX/DY
 S  = ((U(I,J)*U(I,J)-V(I,J)*V(I,J))*PXX+2.0*U(I,J)*V(I,J)*PXY+
&        PHISSP)/(Q2(I,J)/MSQ(I,J)-V(I,J)*V(I,J))
 WPREC = AMAX1(ABS((SIGMA(I,J)-S)/(1.E0+S)),WPREC)
 SIGMA(I,J) = S
 DO 33 I = 2 , NX-1
    DO 33 J = 2 , NY-1
      PHISSP = (U(I,J)*U(I,J)-V(I,J)*V(I,J))*UXP(I,J) + 2.E0*
&                U(I,J)*V(I,J)*UYP(I,J)
      BETA2 = 1.E0 - MSQ(I,J)
      MU = AMAX1(0.E0,BETA2+1.E-10)/(BETA2+1.E-10)
      BE = AMAX1(0.E0,U(I,J))/U(I,J)
      GA = AMAX1(0.E0,V(I,J))/V(I,J)
      MU1 = 1.E0 - MU
      BE1 = 1.E0 - BE
      GA1 = 1.E0 - GA
      CX = U(I,J)/DX
      CY = V(I,J)/DY
      CXY = CX*CY
```

```
          CX = CX*CX
          CY = CY*CY
          A = MU*CY+2.E0*MU1*GA1*((BE-BE1)*CXY-CY)
          B = -2.E0*MU*(CX+CY)+MU1*(4.E0*(BE*(GA-GA1)-GA)*CXY+
     &        CX+2.E0*CXY+CY)
          C = MU*CY+2.E0*MU1*GA*((BE1-BE)*CXY-CY)
          D = MU1*GA1*CY
          E = MU1*GA* CY
          F =-(MU/2.E0+2.E0*MU1*GA1*BE)*CXY
          G = MU*CX+2.E0*MU1*BE*((GA1-GA)*CXY-CX)
          H = (MU/2.E0+2.E0*MU1*BE*GA)*CXY
          II= MU1*BE*CX
          JJ= (MU/2.E0+2.E0*MU1*BE1*GA1)*CXY
          K = MU*CX+2.E0*MU1*BE1*((GA-GA1)*CXY-CX)
          L =-(MU/2.E0+2.E0*MU1*BE1*GA)*CXY
          N = MU1*BE1*CX
          PHISSF = A*PHIF(I,J+1)+B*PHIF(I,J)+C*PHIF(I,J-1)+
     &             D*PHIF(I,J+2)+E*PHIF(I,J-2)+F*PHIF(I-1,J+1)+
     &             G*PHIF(I-1,J)+H*PHIF(I-1,J-1)+II*PHIF(I-2,J)+
     &             JJ*PHIF(I+1,J+1)+K*PHIF(I+1,J)+L*PHIF(I+1,J-1)+
     &             N*PHIF(I+2,J)
          S  = MSQ(I,J)*(PHISSP+PHISSF)*EXT(I,J)/Q2(I,J)
          WPREC = AMAX1(ABS((SIGMA(I,J)-S)/(1.E0+S)),WPREC)
          SIGMA(I,J) = S
   33 CONTINUE
      RETURN
      END
C
C*********************************************************************
C*    SUPERPOSITION DE L'INFLUENCE DES DOUBLETS.                     *
C*********************************************************************
C
      SUBROUTINE SUPERP(NX,NY,MACH,GAMMA,UP,VP,U,V,Q2,MSQ)
      PARAMETER (IMAX=120,JMAX=96)
      REAL*4 U(IMAX,JMAX),V(IMAX,JMAX),UP(IMAX,JMAX),
     &       VP(IMAX,JMAX),Q2(IMAX,JMAX),MSQ(IMAX,JMAX)
      REAL*8 MACH,GAMMA,A2
      INTEGER NX,NY,I,J
      DO 11 I = 1 , NX
        DO 11 J = 1 , NY
          U(I,J) = U(I,J) + UP(I,J)
          V(I,J) = V(I,J) + VP(I,J)
          Q2(I,J) = U(I,J)*U(I,J) + V(I,J)*V(I,J)
          A2 = 1.E0/MACH/MACH+(GAMMA-1.E0)*(1.E0-Q2(I,J))/2.E0
          MSQ(I,J) = Q2(I,J)/A2
   11 CONTINUE
      RETURN
      END
C
C*********************************************************************
```

```
C*    CALCUL DES VITESSES TANGENTIELLES ET DE LA PORTANCE.            *
C*********************************************************************
C
      SUBROUTINE VITPOR(N,CORD,PHI,DS,VIT,CL)
      PARAMETER (IPMAX=121)
      REAL*4 MACH,CL,B,CORD
      REAL*4 DS(IPMAX),PHI(IPMAX),VIT(IPMAX)
      INTEGER I,N
      DO 11 I = 1 , N
         VIT(I) = (PHI(I+1) - PHI(I))/DS(I)
   11 CONTINUE
      CL = 2.E0*(PHI(N+1)-PHI(1))/CORD
      RETURN
      END
C
C*********************************************************************
C*    MISE SUR FICHIER DES RESULTATS POUR PRODUIRE DES GRAPHIQUES.    *
C*********************************************************************
C
      SUBROUTINE WRESUL(N,NX,NY,LX,LY,ARFOIL,ALPHA,MACH,GAMMA,PRECD,
     & CL,X,Y,XC,XG,YG,VIT,U,V,UXP,UYP,SIGMA,ITER,WPREC,CPUSEC,EXT,MSQ)
      PARAMETER (IPMAX=121,ITEMAX=31,IMAX=120,JMAX=96)
      REAL*4 CPUSEC(2)
      REAL*4 X(IPMAX),Y(IPMAX),XC(IPMAX),VIT(IPMAX),WPREC(ITEMAX)
      REAL*4 XG(IMAX),YG(JMAX),U(IMAX,JMAX),V(IMAX,JMAX),
     &       SIGMA(IMAX,JMAX),EXT(IMAX,JMAX),UXP(IMAX,JMAX),
     &       UYP(IMAX,JMAX),MSQ(IMAX,JMAX)
      REAL*4 ALPHA,DX,DY,MACH,CL,GAMMA,CPDO,CPUP,K1,K2,K3,PRECD,LX,LY,
     &       XMIN,XMAX,YMIN,YMAX,PI,RADDEG,X2,Y2,Q2,A2,UMAX,VMAX,SC
      INTEGER I,K,N,ITER,NX,NY
      CHARACTER*10 ARFOIL
      PI = 3.1415926535897932 38E0
      RADDEG = 180.E0/PI
      DX = LX*X(1)/FLOAT(NX)
      DY = LY*X(1)/FLOAT(NY)
      XMIN = XG(1) - DX/2.E0
      XMAX = XG(NX) + DX/2.E0
      YMIN = YG(1) - DY/2.E0
      YMAX = YG(NY) + DY/2.E0
      IF(MACH.NE.0.E0)THEN
        K1 = 2.E0/GAMMA/MACH/MACH
        K2 = (GAMMA-1.E0)*MACH*MACH/2.E0
        K3 = GAMMA/(GAMMA-1.E0)
      END IF
      WRITE(8,20)
      WRITE(8,30) ARFOIL
      WRITE(8,40) ALPHA*RADDEG,MACH,GAMMA
      WRITE(8,50) N
      WRITE(8,60) NX,NY,LX,LY,XMIN,XMAX,YMIN,YMAX
      WRITE(8,110) PRECD
```

```
      WRITE(8,120) (CPUSEC(I),I=1,2)
      DO 11 I = 1,ITER
         WRITE(9,80) I,WPREC(I)
   11 CONTINUE
      DO 22 I = 1 , N
         J = I
         IF (I.GT.N/2) J = N+1-I
         IF (MACH.NE.0.E0) THEN
           CPDO = K1*((1.E0+K2*(1.E0-VIT(I)**2))**K3-1.E0)
C          CPUP = K1*((1.E0+K2*(1.E0-VIT(N+1-I)**2))**K3-1.E0)
         ELSE
           CPDO = 1.E0-VIT(I)*VIT(I)
           CPUP = 1.E0-VIT(N+1-I)*VIT(N+1-I)
         END IF
         WRITE(10,70) 1.E0,XC(J),CPDO
   22 CONTINUE
      IF (MACH.EQ.0.E0) GOTO 99
      K = 1
      DO 33 I = 1 , N+1
         WRITE(11,80) K,X(I),Y(I)
   33 CONTINUE
      K = 2
      UMAX = 0.E0
      VMAX = 0.E0
      DO 44 I = 1 , NX
         DO 44 J = 1 , NY
            WRITE(11,80) K,XG(I),YG(J)
            UMAX = AMAX1(EXT(I,J)*ABS(U(I,J)),UMAX)
            VMAX = AMAX1(EXT(I,J)*ABS(V(I,J)),VMAX)
   44 CONTINUE
      K = 3
      DO 55 I = 1 , NX-1
         WRITE(11,80) K,XG(I)+DX/2.E0,YMIN
         WRITE(11,80) K+1,XG(I)+DX/2.E0,YMAX
   55 CONTINUE
      K = 5
      DO 66 J = 1 , NY-1
         WRITE(11,80) K,XMIN,YG(J)+DY/2.E0
         WRITE(11,80) K+1,XMAX,YG(J)+DY/2.E0
   66 CONTINUE
      K = 0
      SC = 0.8E0*AMIN1(DX/UMAX,DY/VMAX)
      DO 77 J = 1,NY
        DO 77 I = 1,NX
          IF (EXT(I,J).EQ.0.E0) GOTO 77
          K = K + 1
C         Q2 = U(I,J)*U(I,J) + V(I,J)*V(I,J)
C         A2=1.E0/MACH/MACH+(GAMMA-1.E0)*(1.E0-Q2)/2.E0
          X2 = XG(I) + SC*U(I,J)
          Y2 = YG(J) + SC*V(I,J)
```

```
      WRITE(12,70) XG(I),YG(J),SIGMA(I,J),MSQ(I,J)
      WRITE(13,80) K,XG(I),YG(J)
      WRITE(13,80) K,X2,Y2
      WRITE(14,90) I,J,XG(I),YG(J),U(I,J),V(I,J),UXP(I,J),UYP(I,J)
 77 CONTINUE
    WRITE(12,70) XMIN,YMIN,0.0,0.0
    WRITE(12,70) XMAX,YMAX,0.0,0.0
    K = K + 1
    DO 88 I = 1, N+1
       WRITE(13,80) K,X(I),Y(I)
 88 CONTINUE
 99 CONTINUE
 20 FORMAT(1X,'PROGRAMME FIELD PANEL (FORMULATION PAR POTEN. DE VIT.',
   &        ' LINEAIRE)',/)
 30 FORMAT(1X,'PROFIL ',A10)
 40 FORMAT(1X,'   ANGLE D''ATTAQUE          : ',F6.3,/,1X,
   &          '   NOMBRE DE MACH           : ',F6.3,/,1X,
   &          '   GAMMA                    : ',F6.3)
 50 FORMAT(1X,'   NOMBRE DE PANNEAUX        : ',I3,/)
 60 FORMAT(1X,'DOMAINE',/,1X,'   NOMBRE DE DIVISIONS EN X : ',I3,/,
   &       1X,'   NOMBRE DE DIVISIONS EN Y : ',I3,/,
   &       1X,'   LONGUEUR EN X / CORDE    : ',F4.1,/,
   &       1X,'   LONGUEUR EN Y / CORDE    : ',F4.1,/,
   &       1X,'   XMIN                     : ',F10.4,/,
   &       1X,'   XMAX                     : ',F10.4,/,
   &       1X,'   YMIN                     : ',F10.4,/,
   &       1X,'   YMAX                     : ',F10.4,/)
 70 FORMAT(6(E10.3,2X))
 80 FORMAT(I5,6(E12.3,2X))
 90 FORMAT(1X,2(I2,1X),6(E10.3,1X))
110 FORMAT(1X,'PRECISION',19X,': ',D10.3,//)
120 FORMAT(1X,'TEMPS (SEC) POUR EVALUER LES COEF.=',F7.2,
   &       /,1X,'TEMPS (SEC) POUR LES ITERATIONS   =',F7.2)
130 FORMAT(1X,'ITERATION # ',I2,'     PREC = ',D10.3)
    RETURN
    END
```

FIGURES

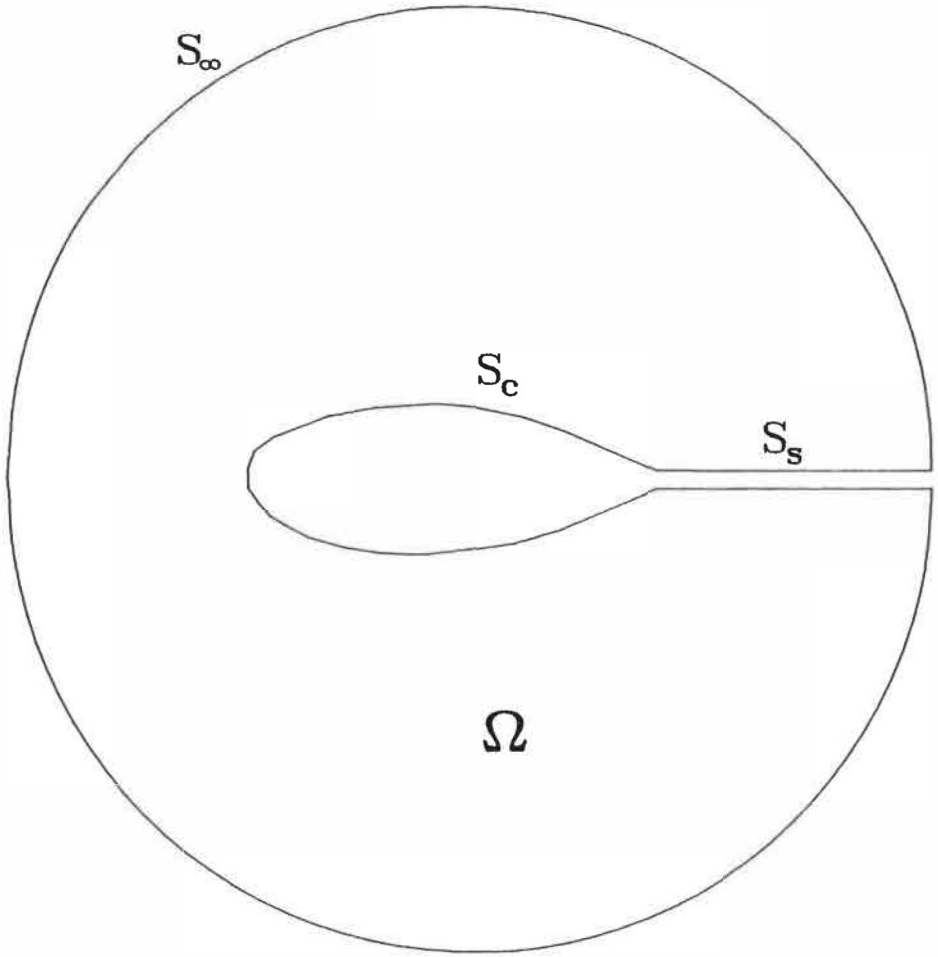

Fig. 1 : Définition du problème

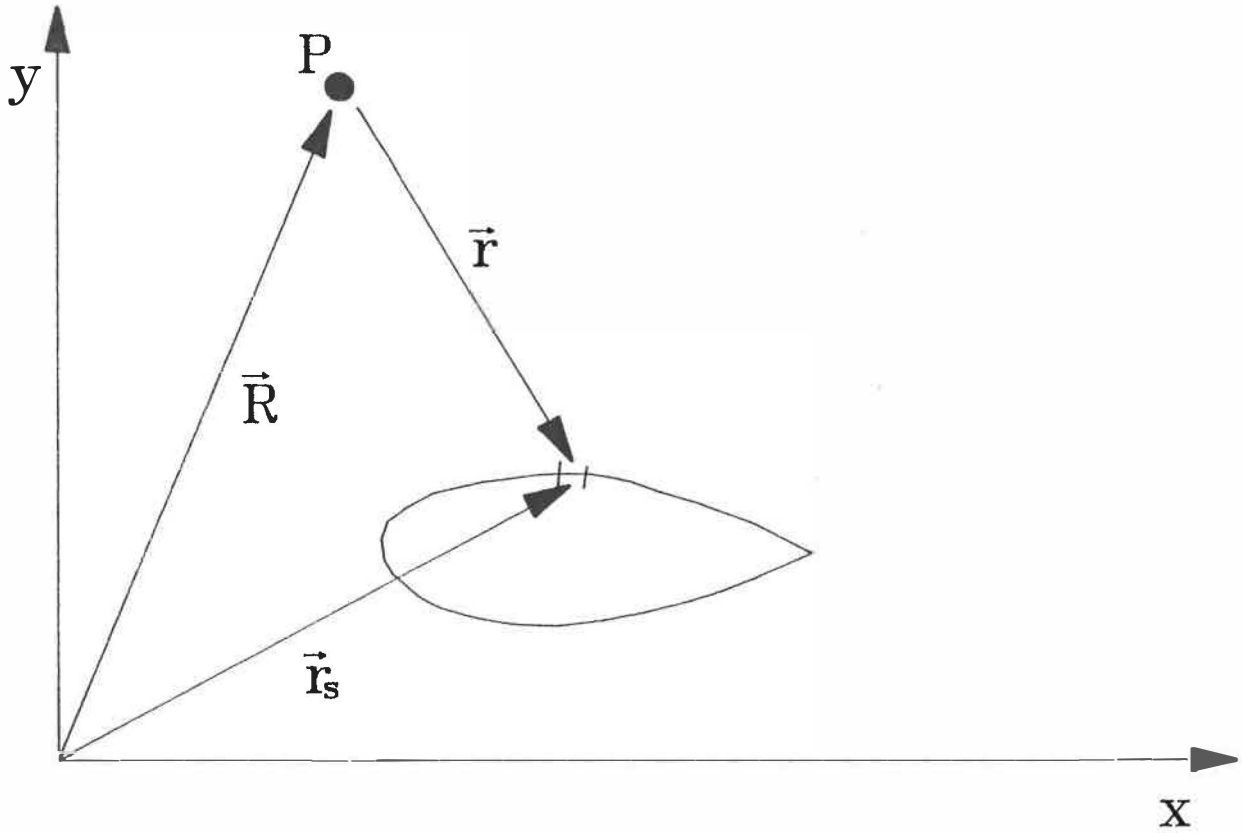

Fig. 2 : Solution intégrale

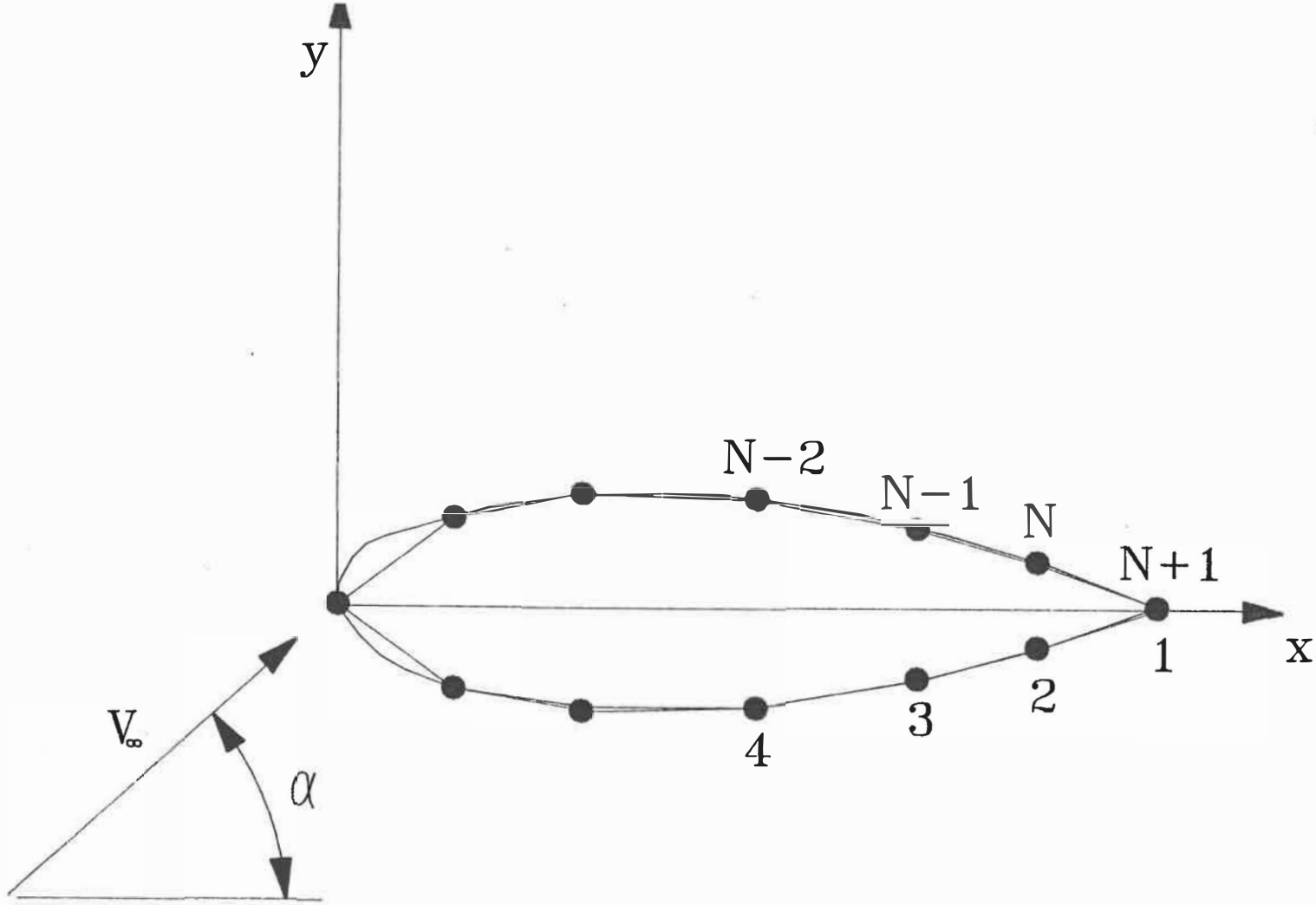

Fig. 3 : Discrétisation du corps

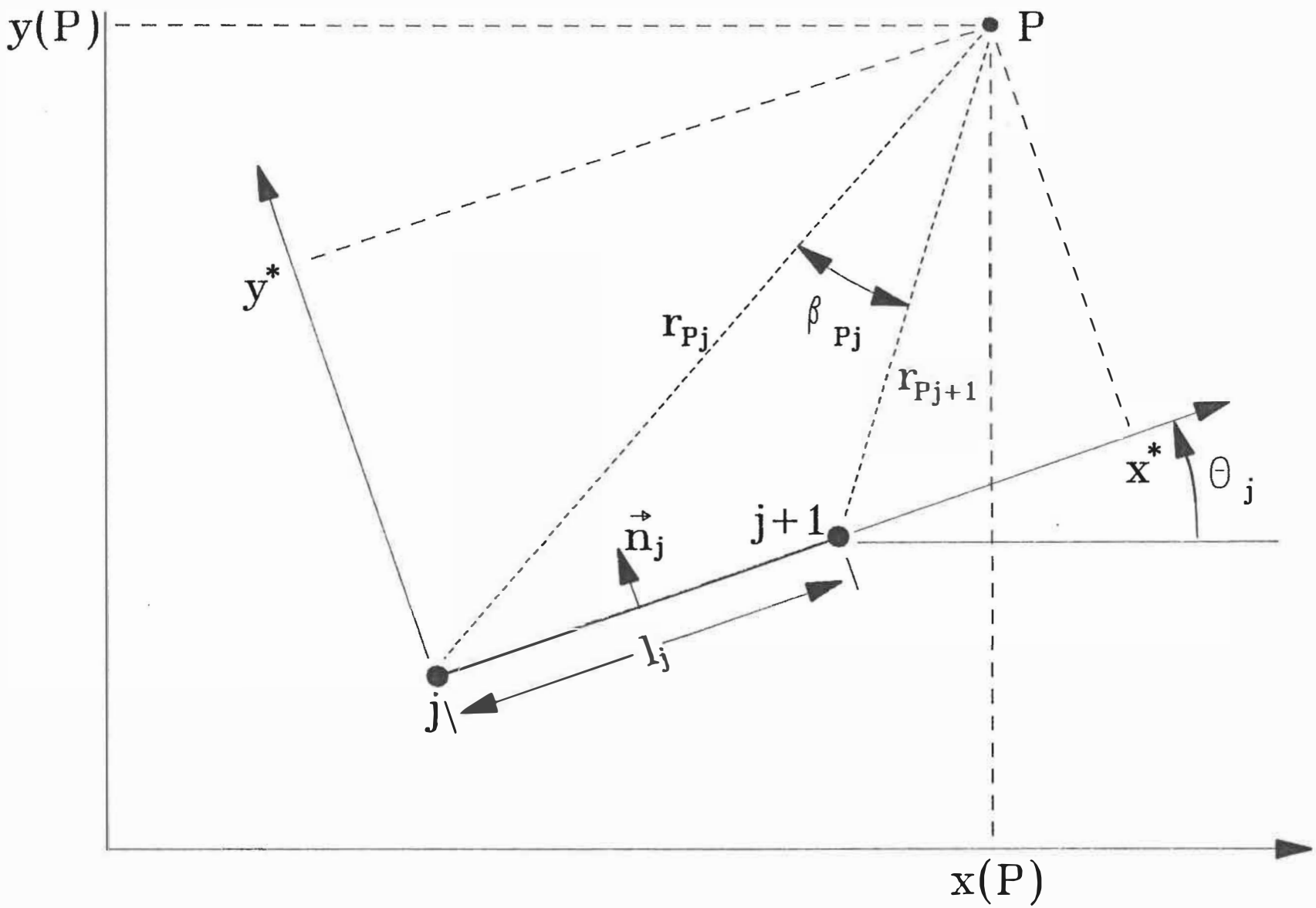

Fig. 4 : Panneau de doublet

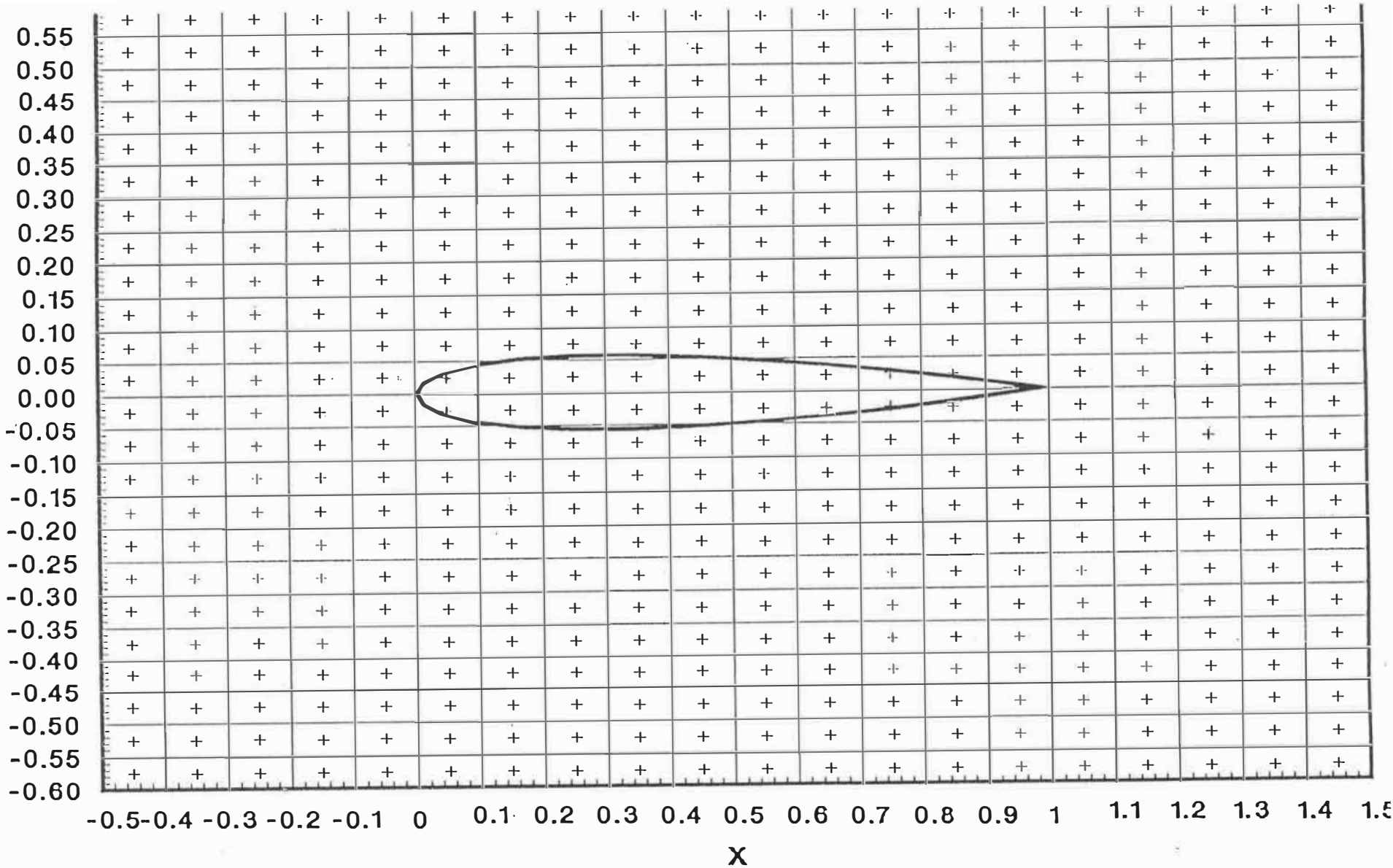

Fig. 5 Domaine de calcul

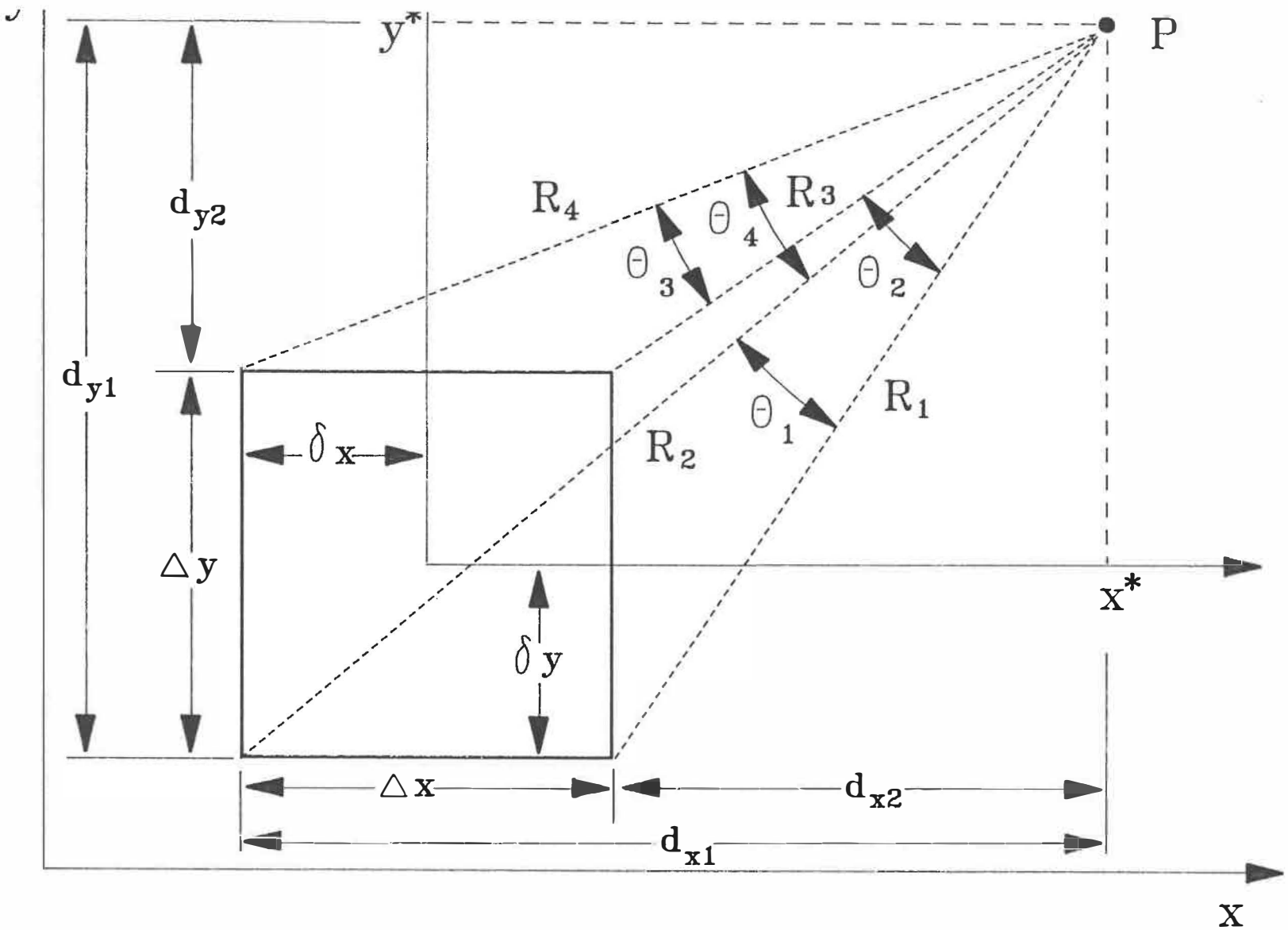

Fig. 6 : Pavé de source

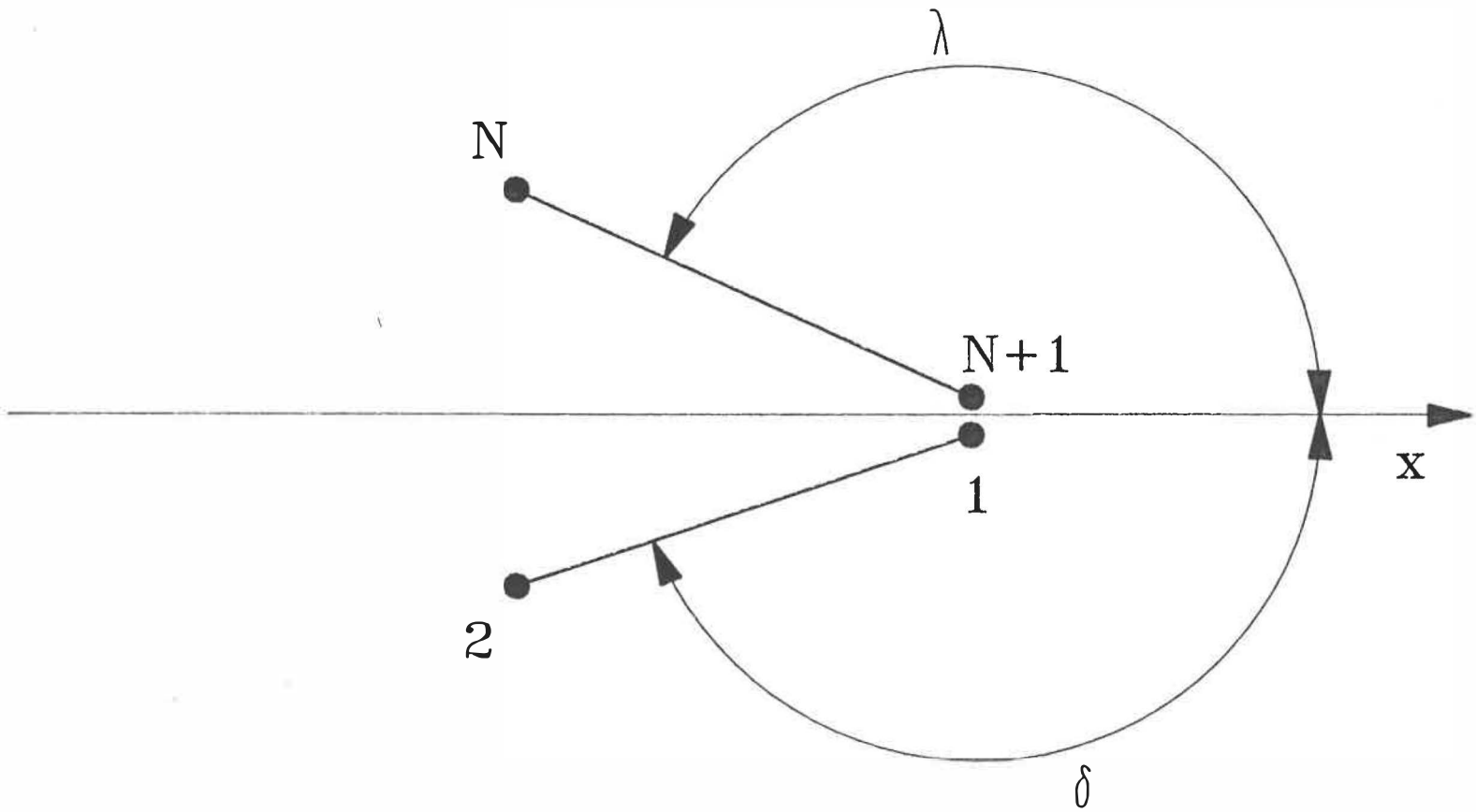

Fig. 7 : Bord de fuite

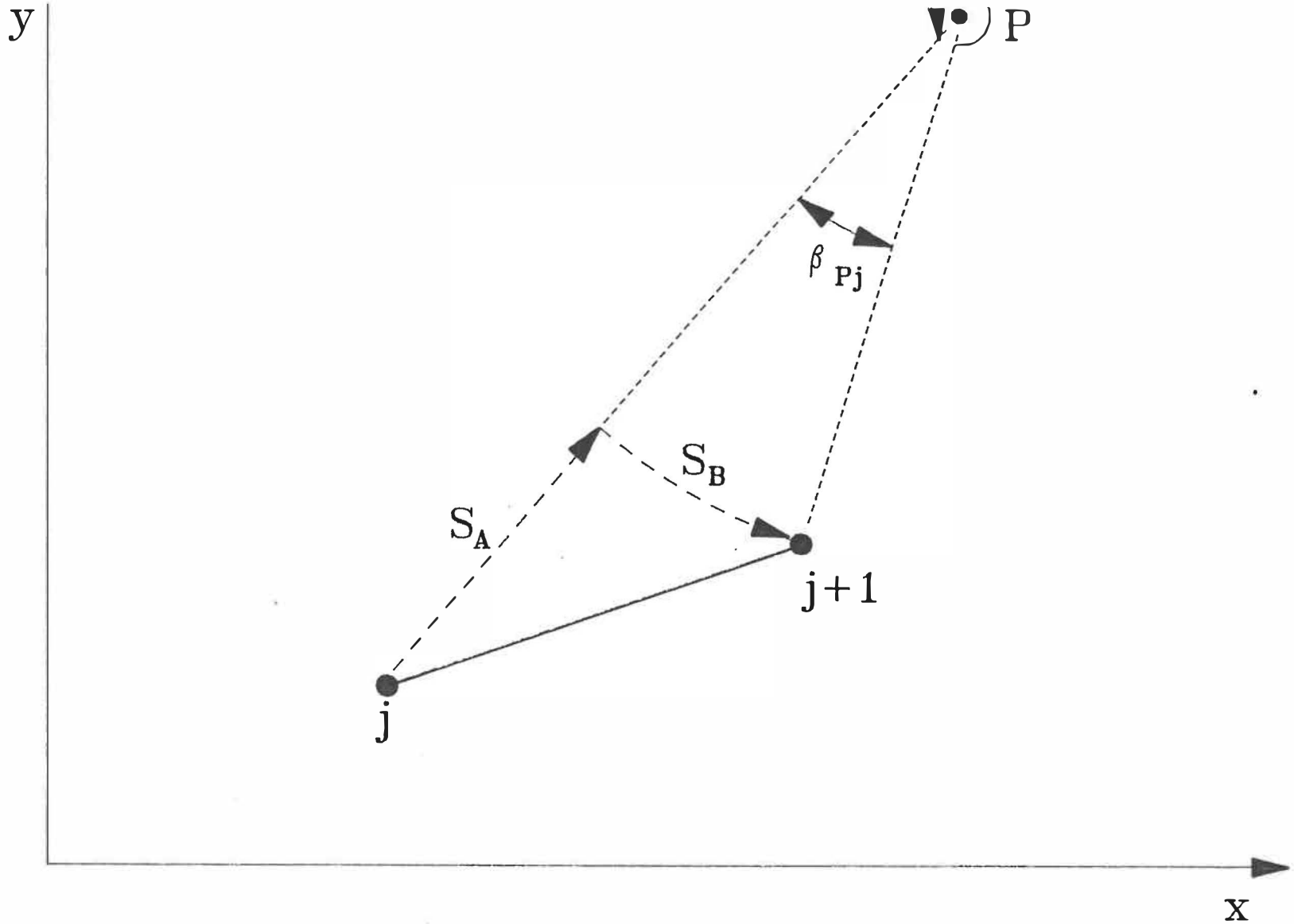

Fig. 8 : Parcours équivalent

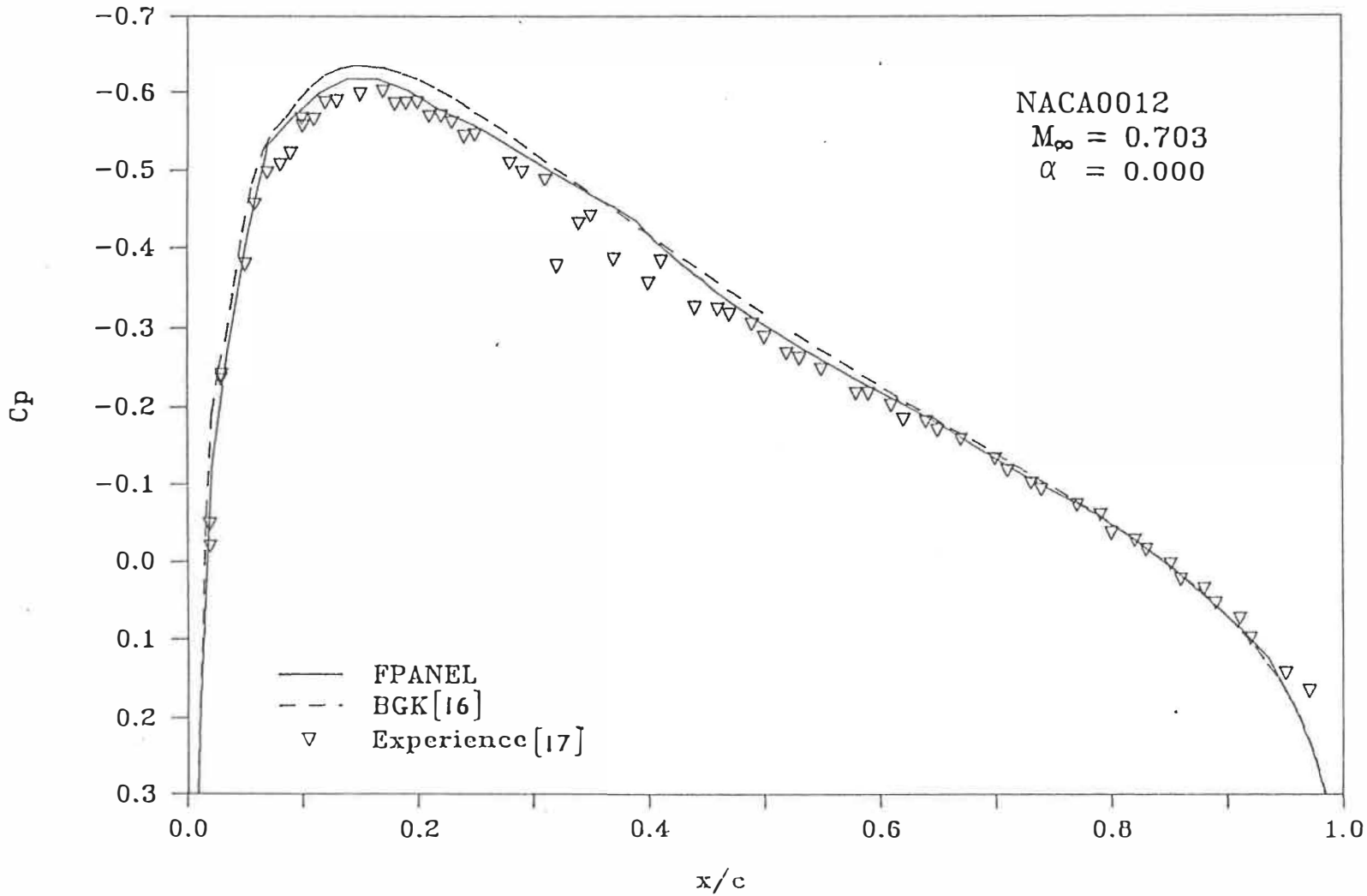

Fig. 9 : Coefficient de pression

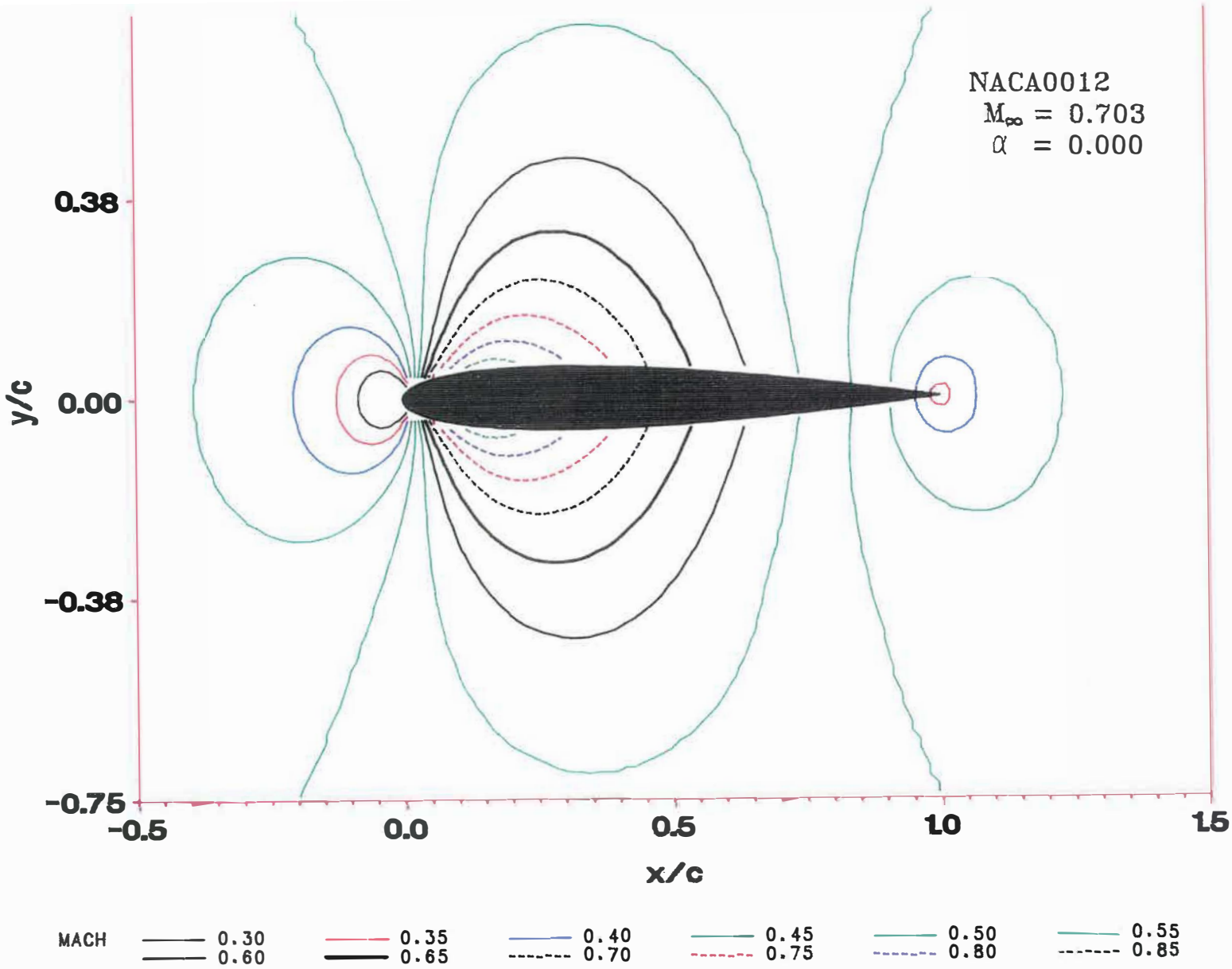

Fig. 10 : Nombre de Mach local

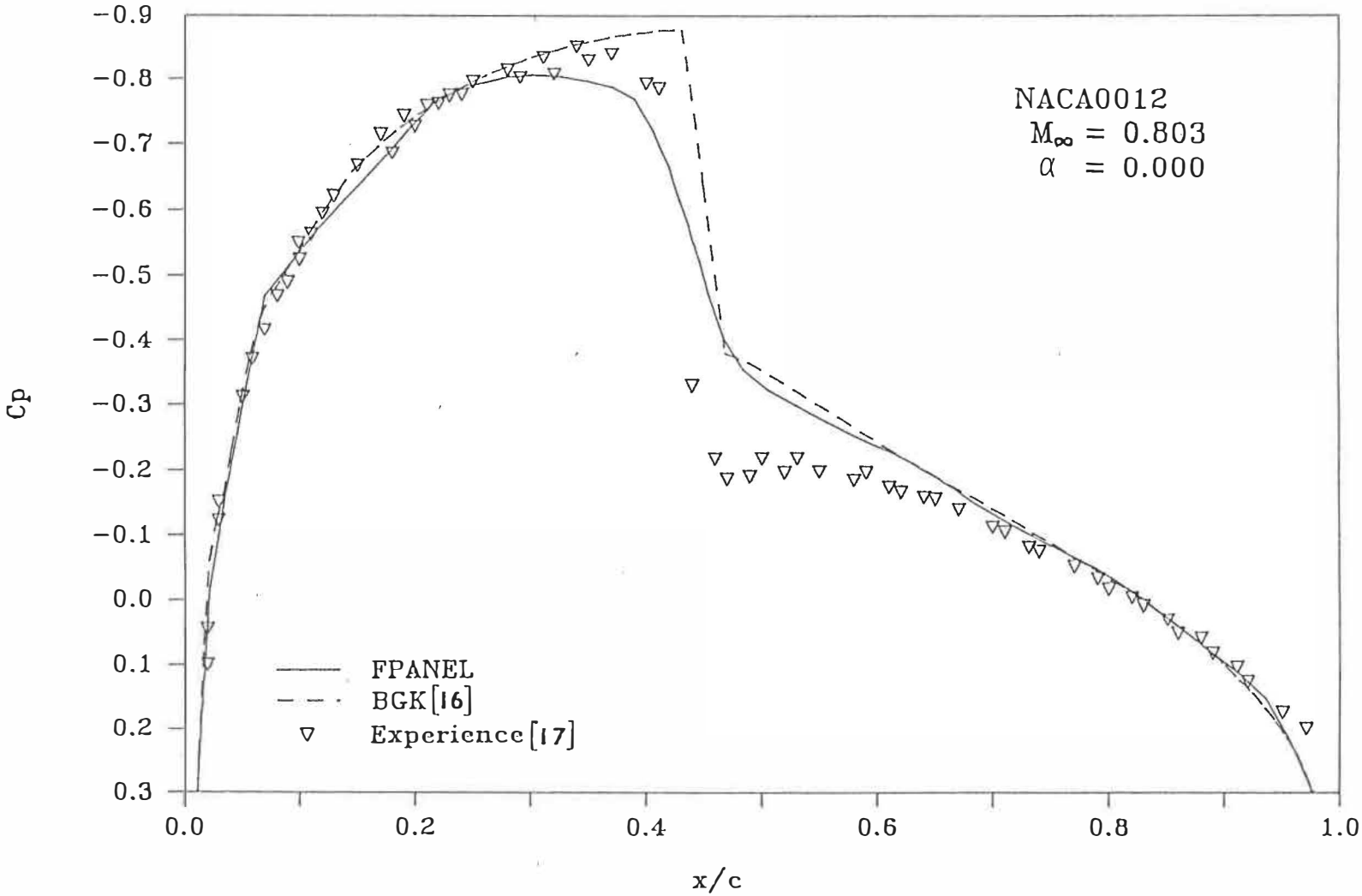

Fig. 11 : Coefficient de pression

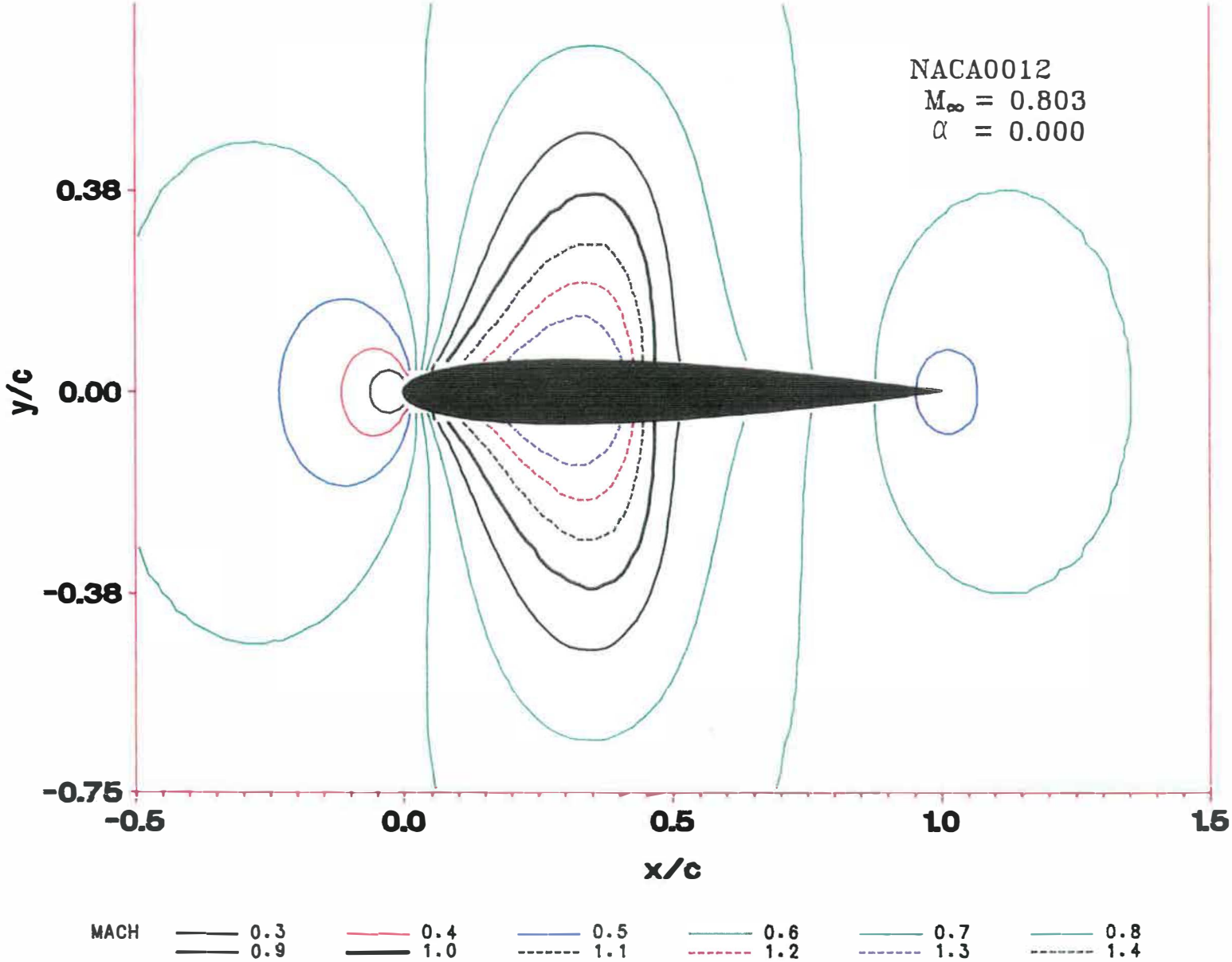

Fig. 12 : Nombre de Mach local

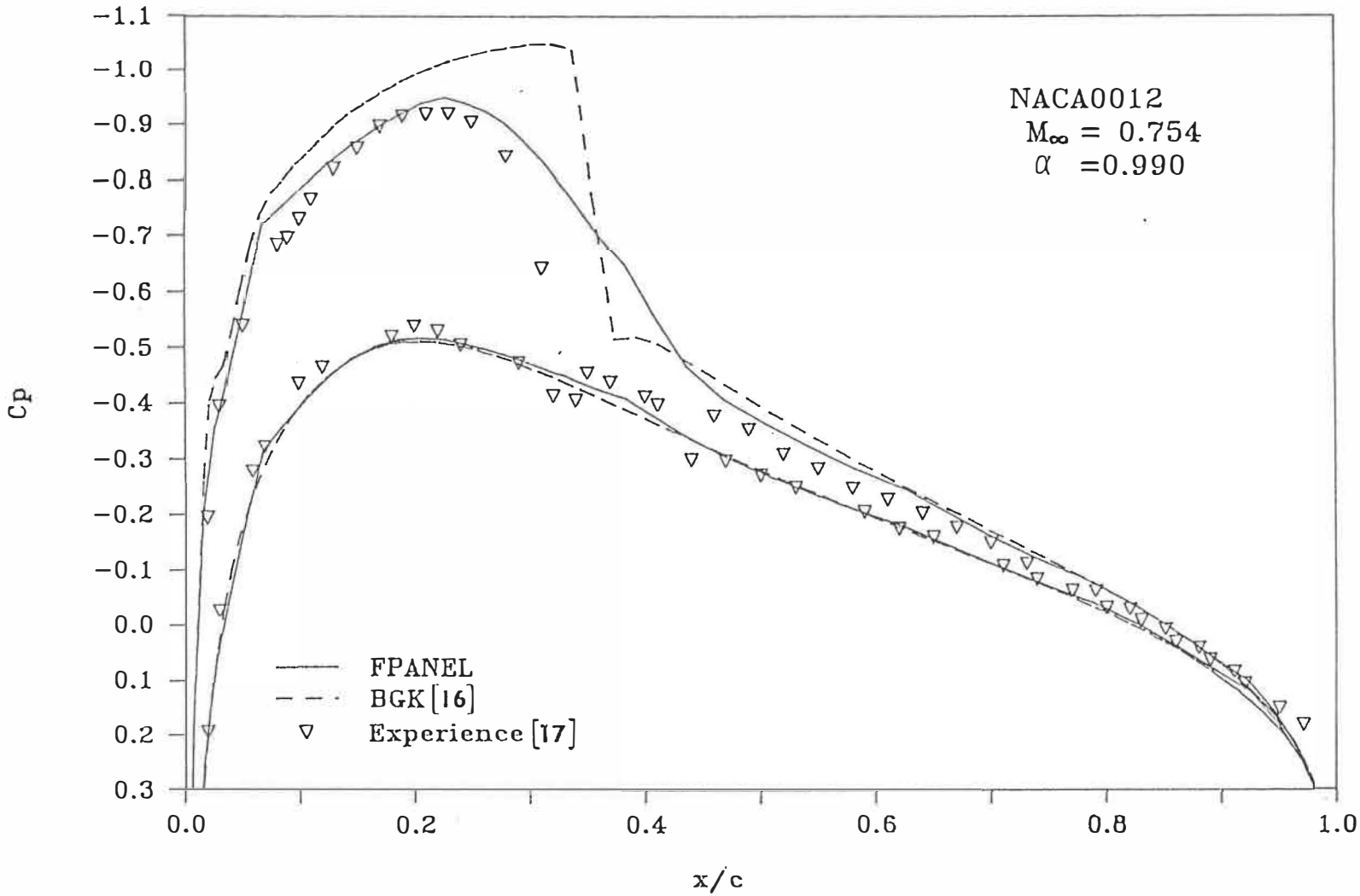

Fig. 13 : Coefficient de pression

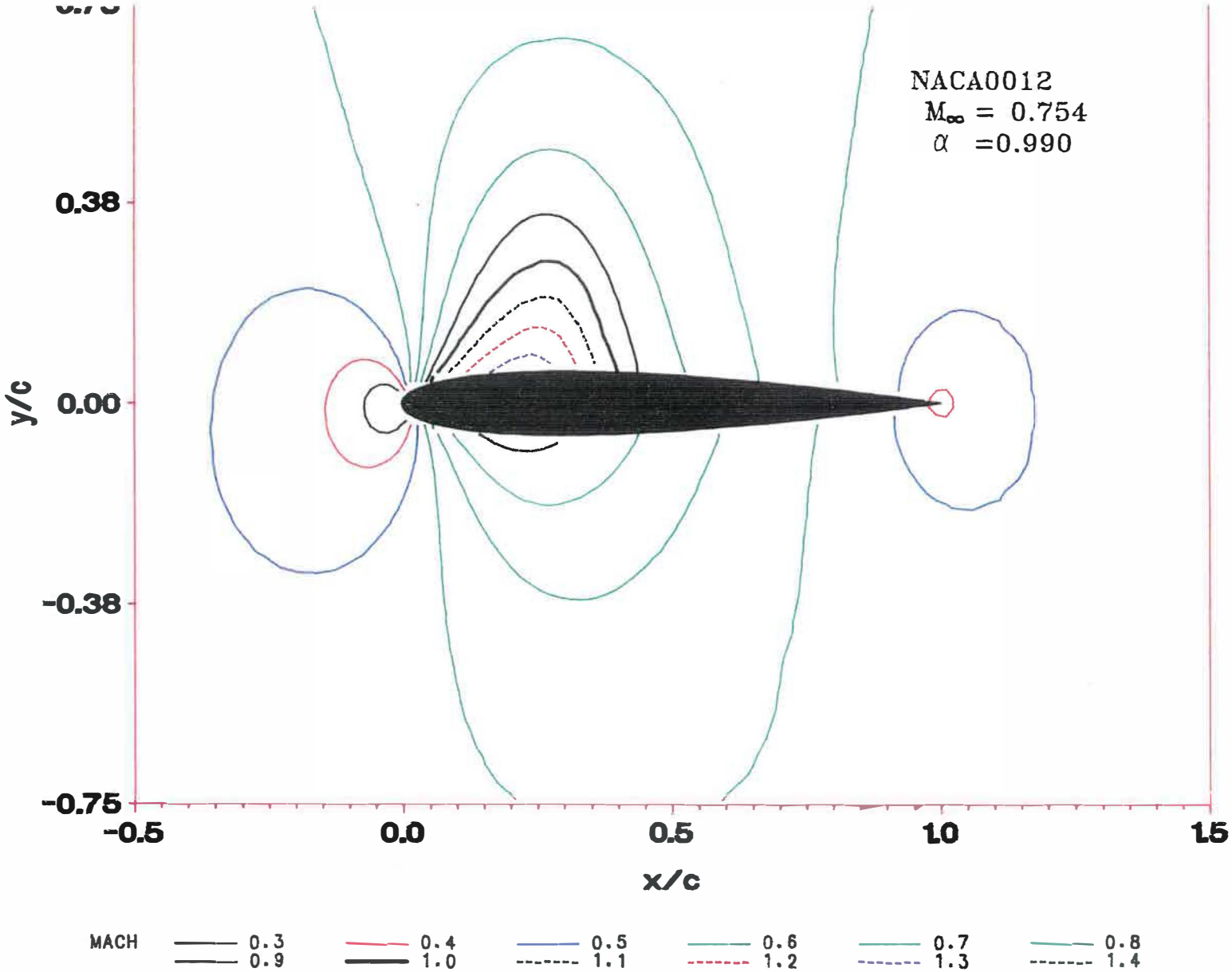

Fig. 14 : Nombre de Mach local

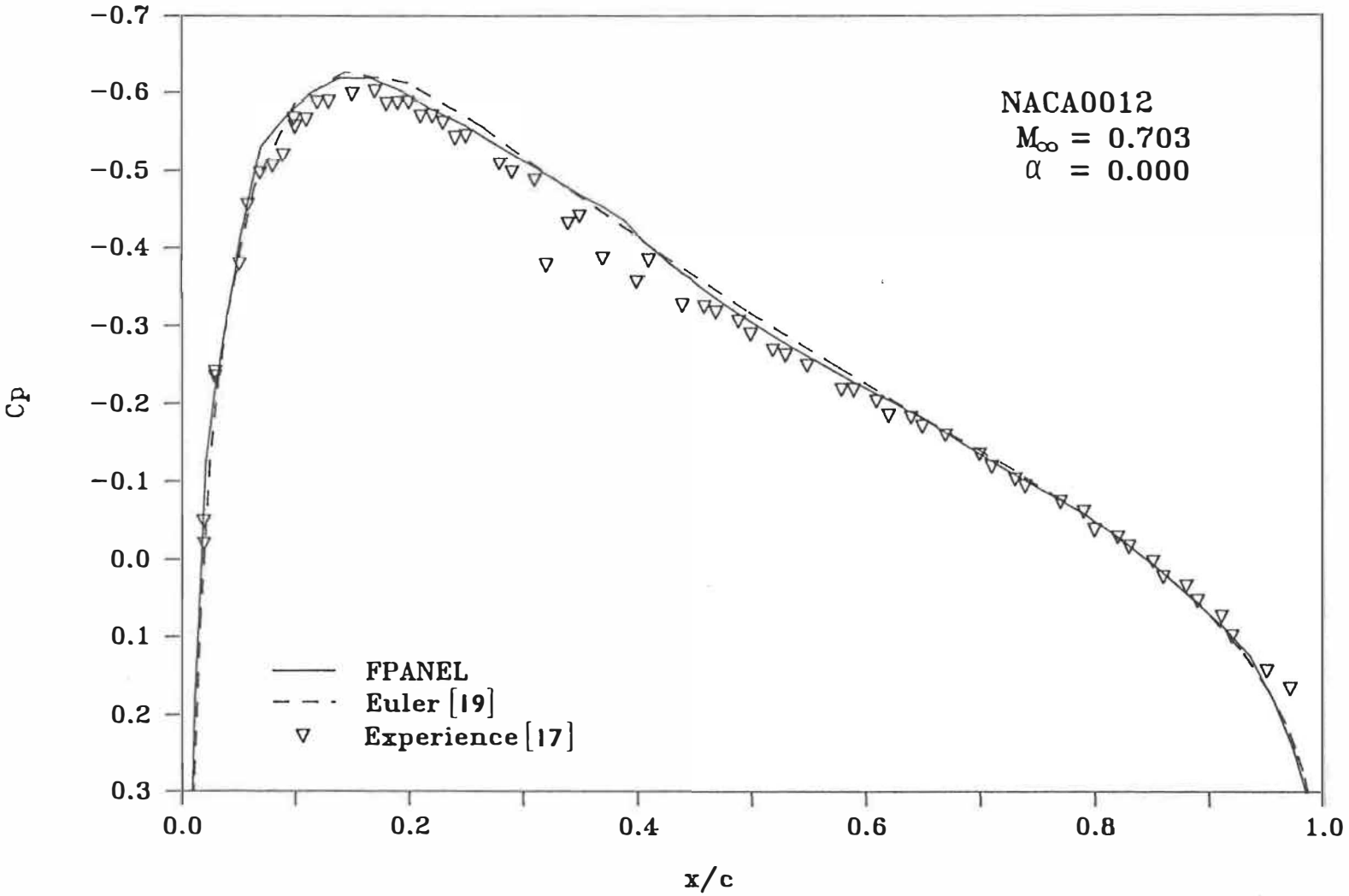

Fig. 15: Coefficient de pression

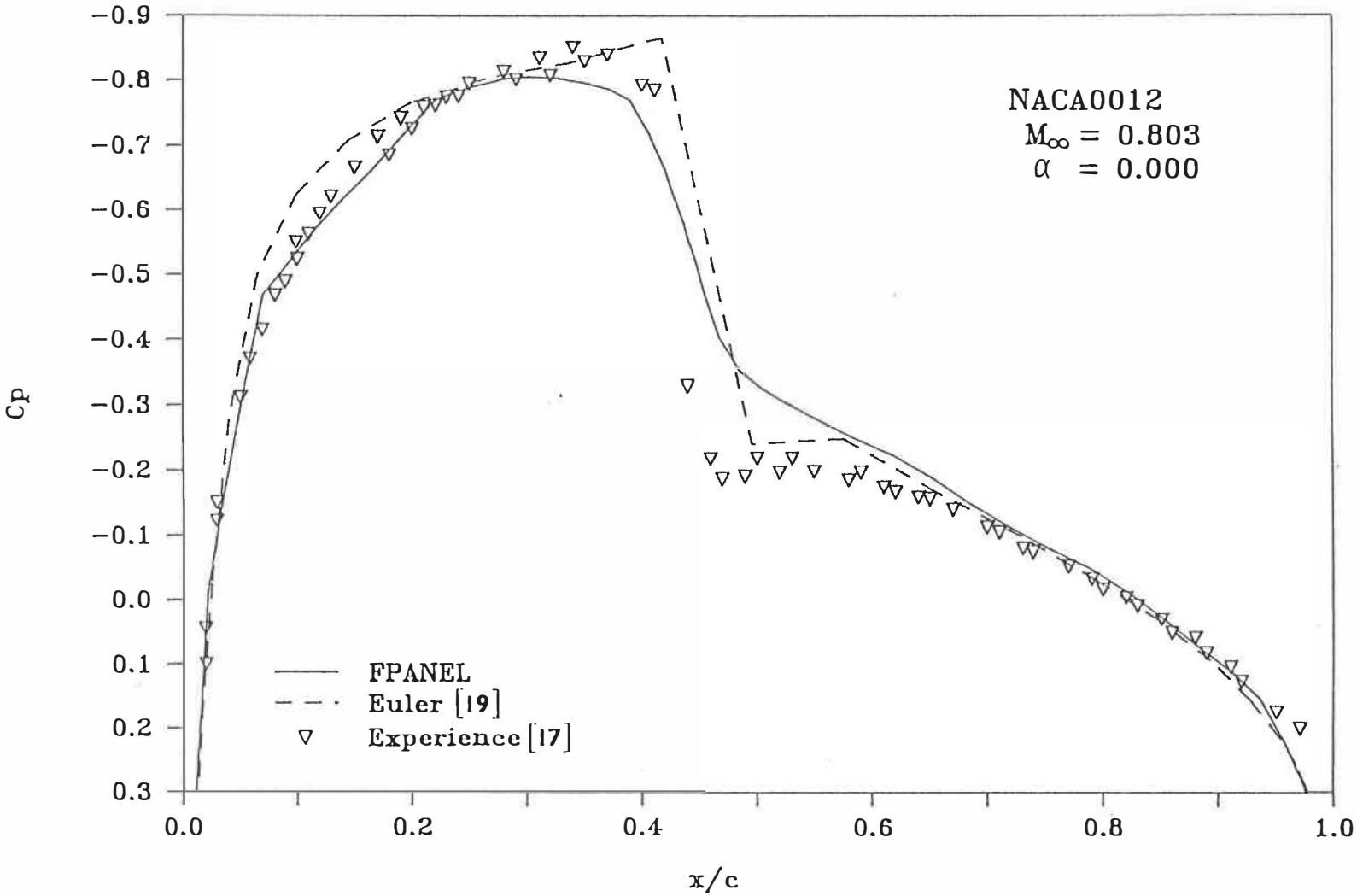

Fig. 16 : Coefficient de pression

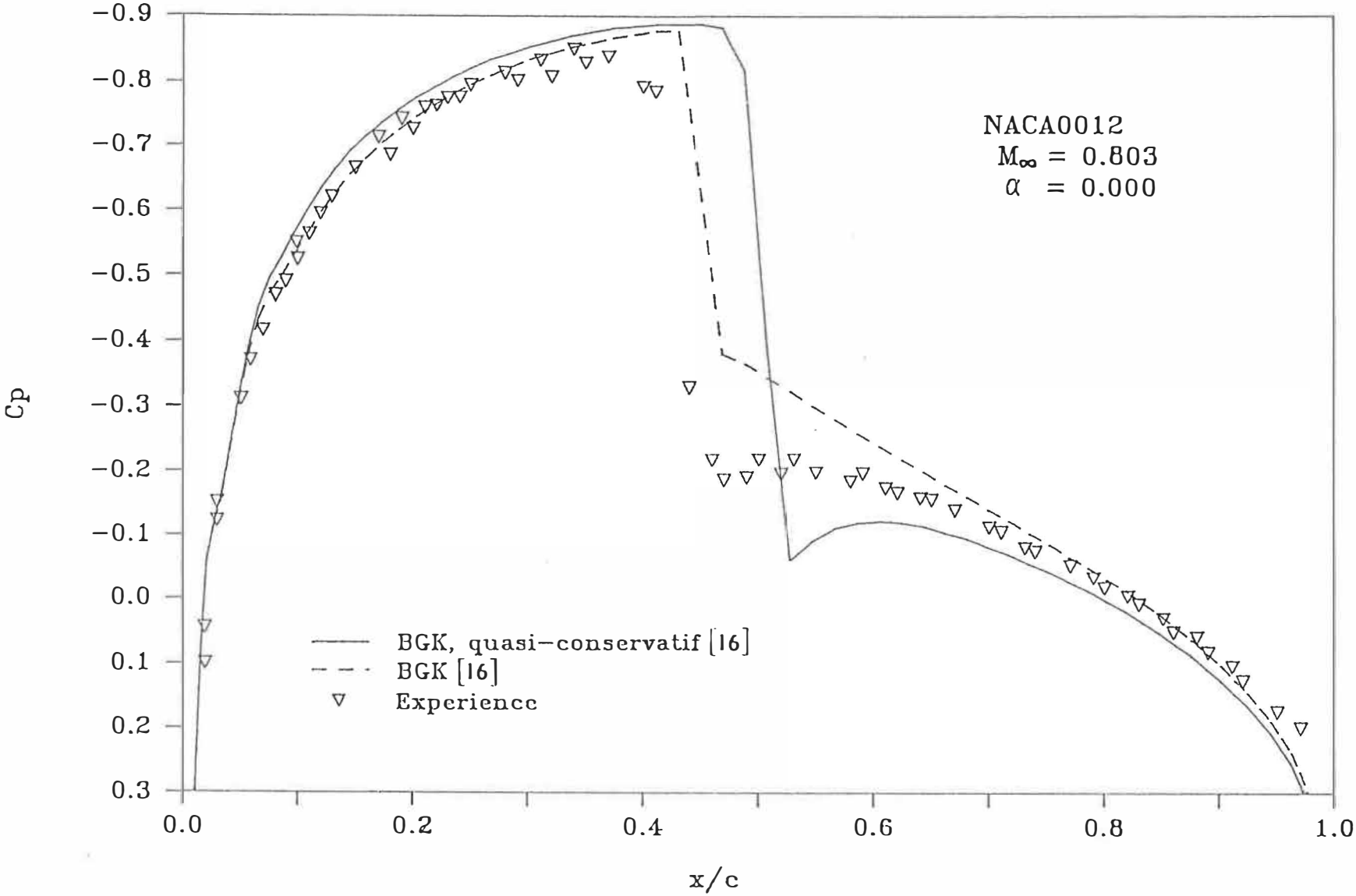

Fig. 17 : Coefficient de pression

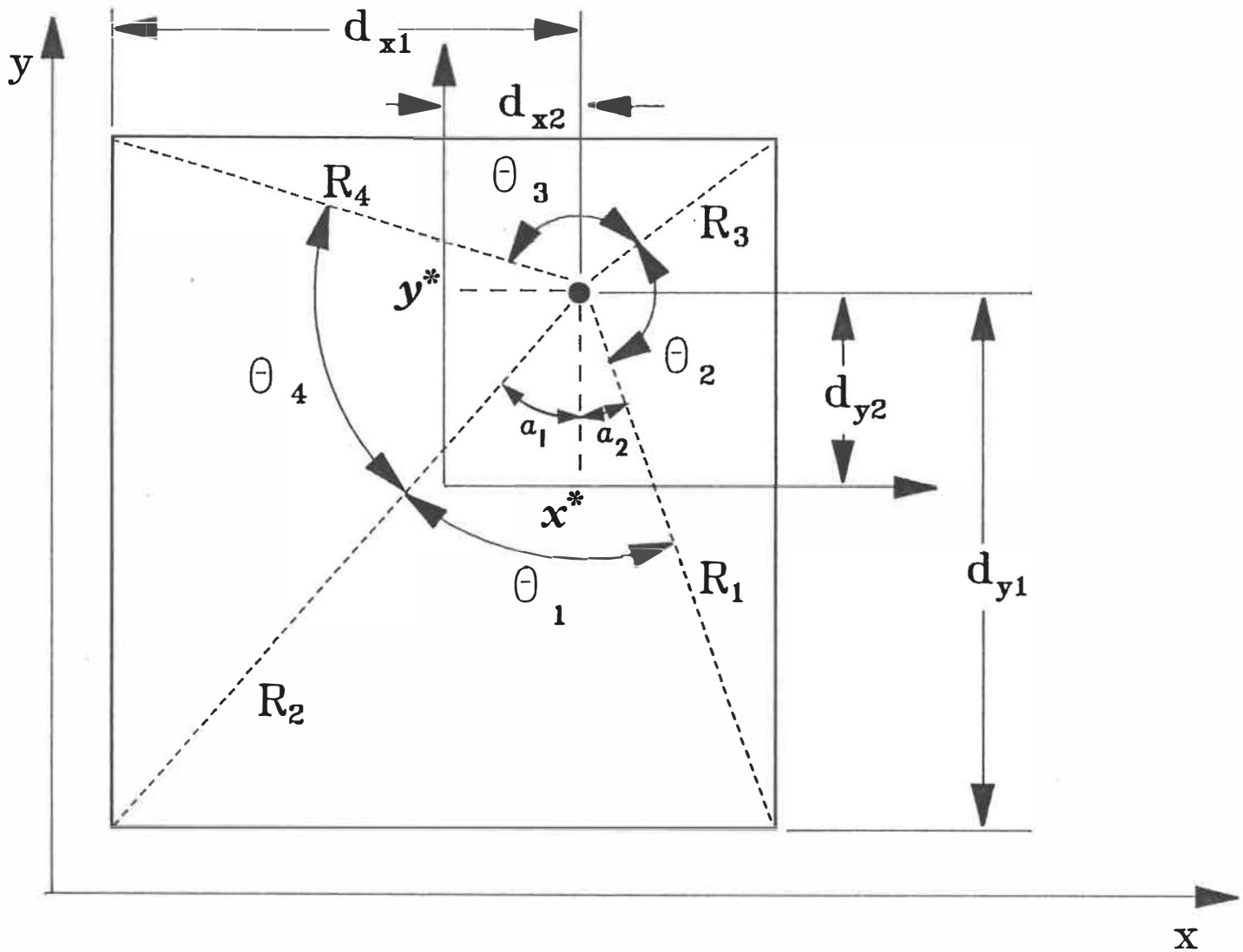

Fig. 18 : Pavé de source, point a l'intérieur du pavé

ÉCOLE POLYTECHNIQUE DE MONTRÉAL
3 9334 00244246 3