

ECOLE POLYTECHNIQUE DE THIES



Projet de fin
d'études

GC. 0254

EN VUE DE L'OBTENTION DU DIPLOME D'INGENIEUR DE CONCEPTION

Titre:

CONCEPTION D'UN LOGICIEL POUR SIMULER PAR
LA METHODE DES ELEMENTS FINIS L'ECOULEMENT
EN MILIEU POREUX SATURE

Auteur:

Ametepe Y. NENONENE

Directeur:

M. Geràrd SOUMA

Co-directeur:

M. Amadou SARR

DATE: JUIN 88

à Papavi Denyo Komi NENONENE

à Papa

à Dada

REMERCIEMENTS

À monsieur Gérard SOUMA professeur,
à monsieur Amadou SARR Professeur,
à monsieur Kangni KINVI,
à tout le personnel du Centre de calcul,
à tous ceux qui de loin ou de près
ont contribué au bon déroulement
de ce travail,
je voudrais adresser mes sincères
remerciements.

SOMMAIRE

L'écoulement en milieu poreux saturé est régi par l'équation de poisson. Une formulation par éléments finis de cette équation permet d'obtenir des systèmes d'équations qui se prêtent à un traitement numérique. Nous nous sommes intéressés dans l'élaboration du logiciel, à l'écoulement permanent. Néanmoins dans les développements mathématiques nous avons intégré la phase transitoire. Le logiciel bien que orienté vers le traitement de l'écoulement hydrodynamique, peut bien s'adapter à la résolution de tout phénomène physique régi par l'équation de poisson.

TABLE DES MATIERES

	REMERCIEMENTS	i
	SOMMAIRE	ii
I	INTRODUCTION	1
II	MODELE PHYSIQUE	4
	2-1 Equation des écoulements en milieu poreux saturé	4
	2-2 Conditions aux limites	7
III	METHODE DES ELEMENTS FINIS	10
	3-1 Idéatisation et choix des éléments	11
	3-2 Choix des fonctions de pondération	12
	3-3 Introduction des éléments de référence.	14
IV	METHODE NUMERIQUE	17
	4-1 Formulation intégrale.	17
	4-2 Discrétisation de la forme intégrale	19
	4-3 Choix des fonctions de pondération	21
	4-4 Discrétisation du domaine et formulation matricielle	22

	4-5	Introduction des éléments de référence	25
	4-6	Modèle 1-0, 2-0 dans un espace 3-0	27
	4-7	Méthode d'intégration numérique	30
V		ASSEMBLAGE ET RESOLUTION	34
	5-1	Présentation de la méthode frontale	35
	5-2	Exemple de calcul	38
VI		PRESENTATION DU LOGICIEL	45
	6-1	Organigramme du programme	45
	6-2	Lecture des données	46
	6-3	Traitement d'un élément	48
	6-4	Précision des calculs	50
	6-5	Tests de vérification	52
VII		CONCLUSION	-
VIII		REFERENCES	
		Les caractéristiques des éléments de références.	53
		Nomenclature	56
		Résultat des tests	60
		Listings du programme	70
		Bibliographie	131

I

INTRODUCTION

Aucun pays au monde n'est épargné du problème de déficit hydrique. Mais ce dernier se pose encore avec plus d'acuité dans les pays de l'Afrique subsaharienne.

Face à ce fléau, une bonne gestion des ressources hydriques s'impose. On ne doit donc plus se contenter d'une évaluation grossière des réserves en eau souterraine, réserves qui constituent une partie essentielle du bilan hydraulique complet.

Pour répondre à cette exigence, différents modèles mathématiques et numériques ont été développés parmi lesquels :

- La méthode des différences finies
- La méthode des volumes finis
- La méthode des éléments finis

De nos jours la méthode des éléments finis est la plus répandue pour sa grande malléabilité et sa bonne précision. Elle fut proposée par

Turner, Clough, Martin et Topp en 1956.

Si les premières applications de la méthode des éléments finis ont été orientées vers la résolution des équations d'équilibre en élasticité ou en élastoplasticité, il est apparu que cette méthode pouvait s'adapter à la résolution des problèmes physiques régis par des systèmes d'équations aux dérivées partielles.

En particulier, les problèmes se ramenant à la résolution des équations de Laplace ou de Poisson tel que :

- les problèmes de répartition de la charge hydraulique ou des débits dans les écoulements en milieu poreux,

- les problèmes de répartition de température ou de potentiel électrique, peuvent être traités aisément par la méthode des éléments finis.

Après avoir établi le modèle physique du problème, nous passerons à une présentation générale de la mé-

thode des éléments finis. Nous aborderons ensuite la formulation par éléments finis. Nous pourrions à ce stade procéder à l'élaboration du logiciel qui sera écrit en langage turbo-pascal. Dans l'analyse de ce dernier nous discuterons entre autres des problèmes relatifs à la précision des calculs.

II

MODELE PHYSIQUE

Tout problème d'écoulement du fluide newtonien (fluide dont la tension de cisaillement est une forme linéaire du gradient de vitesse) se ramène à la détermination de six inconnues :

ρ : la masse volumique du fluide

p : la pression

θ : la température

v_x, v_y, v_z les composantes du champ de vitesse.

Ces variables pouvant être ou non fonction des variables spatio-temporelles.

Nous nous proposons d'établir une équation qui lie ces inconnues dans le cas particulier d'un écoulement en milieu poreux saturé.

2.1 Équation des écoulements en milieu poreux saturé :

Nous n'allons pas nous apercevoir sur les développements mathématiques qui permettent d'établir l'équation. (Nous invitons le lecteur à se référer à la ré-

férence (4) pour plus de détails).

Pour établir l'équation générale qui régit l'écoulement en milieu poreux saturé, nous disposons des trois équations suivantes :

- Equation de continuité du fluide :

$$\operatorname{div}(\rho \vec{v}) + \frac{\partial}{\partial t}(\rho m) + \rho g = 0 \quad (1)$$

- Equation de continuité du solide :

$$\operatorname{div}[\rho_s(1-m)\vec{v}_s] + \frac{\partial}{\partial t}[(1-m)\rho_s] = 0 \quad (2)$$

- loi de Darcy :

$$\vec{v} - m\vec{v}_s = -\frac{k}{\mu} [\operatorname{grad} p + \rho g \operatorname{grad} z] \quad (3)$$

avec :

ρ : masse volumique du fluide (kg/m^3)

m : porosité du milieu.

g : débit volumique (m^3/s)

\vec{v} : vitesse apparente du fluide (m/s)

\vec{v}_s : vitesse réelle moyenne du solide (m/s)

ρ_s : masse volumique du solide (kg/m^3)

p : pression ($\text{kg}/\text{m}\cdot\text{s}^2$)

z : cote fixant l'altitude par rapport à un référentiel donné. (m)

g : accélération de la pesanteur

\bar{k} : perméabilité intrinsèque

μ : viscosité dynamique

Ces équations sont écrites en coordonnées d'Euler. Si nous portons dans

(1) la valeur de \vec{v} donnée par (3) et si on fait intervenir la notion de

dérivée particulière soit $(\frac{dF}{dt} = \frac{\partial F}{\partial t} + \vec{v}_i \text{grad}_i F)$

en suivant le solide, on obtient:

$$\text{div} \left[p \frac{\bar{k}}{\mu} (\text{grad} p + p g \text{grad} z) \right]$$

$$= m \frac{dp}{dt} + \frac{p}{1-m} \frac{dm}{dt} - \frac{f m}{p_s} \frac{dp_s}{dt} + p g \quad (4)$$

en tenant compte de l'équation d'état du solide et du liquide, en faisant l'hypothèse que les contraintes effective et totale sont constantes et que p est

peu variable dans l'espace, en supposant enfin la relation $p g \frac{\partial h}{\partial t} \approx \frac{\partial p}{\partial t}$,

l'équation (4) devient:

$$\text{div} \bar{k} (\text{grad} h) = p m g \left(\beta_f - \beta_s + \frac{\alpha}{m} \right) \frac{\partial h}{\partial t} + g$$

avec

$\bar{k} = \rho g \frac{\bar{k}}{\mu}$: tenseur de perméabilité
 β_f : compressibilité du fluide
 β_s : compressibilité des grains solides
 φ : compressibilité du milieu poreux
 $h = \frac{P}{\rho g} + z$: charge hydraulique.

avec la notation :

$$C_s = \rho m g (\beta_f - \beta_s \frac{\alpha}{m}) \quad \text{nous obtenons}$$

l'équation générale de l'écoulement en milieu poreux saturé qui s'écrit sous la forme :

$$\text{div} (\bar{k} \text{ grad } h) = C_s \frac{\partial h}{\partial t} + \varphi \quad (5)$$

C_s : coefficient d'emmagasinement spécifique.

2.2 Conditions aux limites et conditions initiales :

À l'équation (5) définie dans un domaine V , on associe trois types de conditions aux limites sur le con

tour S .

1 - Condition de flux imposé ou condition de Neuman

$$-K \frac{\partial h}{\partial n} = q_s(p) \text{ sur le contour } S_1$$

2 - Condition de potentiel imposé ou condition de Dirichlet

$$h = \bar{h}(p) \text{ sur le contour } S_2$$

3 - Condition mixte ou de Cauchy :

$$K \frac{\partial h}{\partial n} = E(p)(h - h_e) \text{ sur le contour } S_3$$

avec : $S_1 \cup S_2 \cup S_3 = S$

$\bar{h}(p)$: potentiel imposé au point P .

$q_s(p)$: flux imposé au point P

$E(p)$: Coefficient d'échange au point P

h_e : potentiel extérieur

n : normale à la surface dirigée positivement vers l'extérieur.

A ces conditions, on ajoute, pour des problèmes transitoires, des conditions initiales qui sont généralement la donnée d'une charge à un temps $t = 0$.

Nous allons passer à la présentation de la méthode des éléments finis que nous appliquerons par la suite à la résolution de l'équation (5).

III

METHODE DES ELEMENTS FINIS

L'ingénieur dans son travail est appelé à résoudre des problèmes très complexes. Il fait donc souvent appel à des méthodes numériques de résolution. L'une des méthodes, utilisées de nos jours pour la résolution des systèmes d'équations aux dérivées partielles est la méthode des éléments finis.

L'approche par éléments finis comprend deux principales étapes :

- La première est une discrétisation du milieu qui consiste à subdiviser le milieu continu (domaine de travail) en une série de sous domaines appelés éléments.

- La deuxième étape est une autre discrétisation de la fonction inconnue (dans notre cas la charge H). Elle consiste à remplacer la valeur exacte de H soit H_{ex} , par une valeur approchée H_{ap} . La valeur de la fonction inconnue H_i en un point (x) d'un élé-

ment sera fonction des coordonnées de ce point et des valeurs $\{H_n\}$ prises par la fonction aux nœuds de l'élément. On cherchera à avoir $(H_{\text{ex}} - H_{\text{op}})$ suffisamment petit.

On procédera ensuite à l'assemblage des équations traduisant le comportement de chaque élément. On obtient ainsi une équation générale linéaire ou non du système global.

Nous allons exposer de façon explicite, les étapes à suivre dans la résolution d'un problème.

3.1 Idéalisation et choix des éléments:

On subdivise comme nous l'avons mentionné le domaine en sous-domaines. Chaque élément doit être défini de façon unique en fonction des coordonnées des nœuds qui le compose.

La partition du domaine doit respecter deux règles.

1. Deux éléments distincts ne doivent avoir en commun que des points situés sur leur frontière commune.

2. l'ensemble de tous les éléments doit constituer un domaine aussi proche que possible du domaine d'étude.

les éléments à une dimension peuvent être linéaire, quadratique ou cubique.

les éléments à deux dimensions sont des triangles ou quadrilatères dont les côtés sont représentés par des courbes polynomiales du 1^{er}, 2^{es} ou 3^{es} degré.

les éléments à trois dimensions sont représentés par des tétraèdres, hexaèdres ou prismes dont les faces sont des surfaces polynomiales du 1^{er}, 2^{es} ou 3^{es} degré. (Nous présenterons dans l'annexe les caractéristiques des éléments utilisés dans notre étude).

3.2 CHOIX DES FONCTIONS D'INTERPOLATION

la variable exacte cherchée est approximée par une variable H_{ap} définie par :

$$H_{ap} = \langle N_1(x), N_2(x), \dots, N_n(x) \rangle \begin{Bmatrix} H_1 \\ H_2 \\ \vdots \\ H_n \end{Bmatrix}$$

que nous notons :

$$H_{ap} = \langle N(x) \rangle \{H_n\}$$

x : point quelconque d'un élément du domaine.

n : Nombre de nœuds de l'élément

$\{H_n\}$: vecteur colonne composé des valeurs de la variable nodale aux nœuds de l'élément.

N_i : les fonctions d'interpolations.

Les fonctions d'interpolations sont choisies de façon à faciliter les calculs numériques.

Puisque nous devons avoir $H_{ap}(x_i) = H_i$ donc les fonctions d'interpolation doivent vérifier :

$$N_j(x_i) = \begin{cases} 0 & \text{si } i \neq j \\ 1 & \text{si } i = j \end{cases}$$

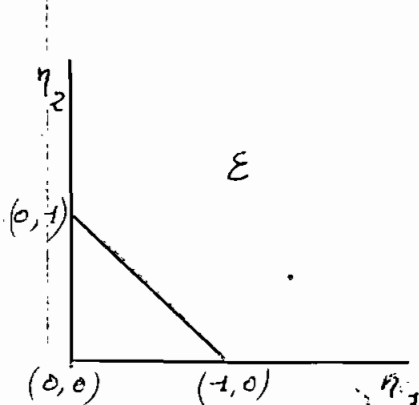
La fonction approchée H_{ap} doit être continue sur les éléments et entre les éléments. Il faut donc que les fonctions $N_i(x)$ soient continues sur les éléments et entre les éléments. Dans certaines con-

ditions il est nécessaire que les fonctions $N_i(x)$ aient des dérivées continues jusqu'à l'ordre $s-1$ (s étant l'ordre de dérivabilité maximale)

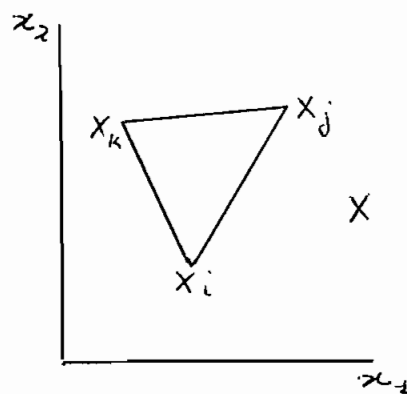
Pour la conception de notre logiciel, nous avons utilisé des fonctions d'interpolation déjà construites (Nous présenterons un tableau de ces fonctions dans l'annexe)

3.3 INTRODUCTION DES ELEMENTS DE REFERENCE

Ce concept permet d'une part de faciliter largement le traitement numérique du problème, d'autre part de simplifier l'expression mathématique à intégrer. Il permet, à travers une transformation géométrique, de lier tous les éléments de même type à un unique élément de référence. Considérons les éléments suivants:



Elément de référence



Elément réel

ξ_i est un point de l'élément de référence
 x_i un point de l'élément réel.

On définit une application ξ de l'espace
de référence E vers l'espace réel X par

$$\xi: E \rightarrow X = X^e(E) = X^e(E, x_i, x_j, x_k) \dots$$

L'application ξ doit être bijective. De plus,
l'image d'un nœud de l'élément de réfé-
rence doit être un nœud de l'élément
réel et une frontière de l'élément réel
doit être l'image d'une frontière de
l'élément de référence.

En posant

$$X^e(E) = \langle \bar{N}_1(E), \bar{N}_2(E) \dots \bar{N}_n(E) \rangle \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix}$$

on peut choisir les fonctions

\bar{N}_i tel que les conditions précitées soient
satisfaites.

Les fonctions \bar{N}_i sont des fonctions d'inter-
polation géométriques. Elles ont les mêmes
propriétés que les fonctions d'interpolation
dans l'approximation de la variable nodale.

Si on établit une bijection ξ entre un élément
réel et un élément de référence tel que $\xi(E) = X$

alors on démontre que :

$$H_{ap} = \langle N(x) \rangle \{H_n\} = \langle N(\epsilon) \rangle \{H_n\}$$

Dans nos calculs nous choisissons $\bar{N} \equiv N$.
On remarque que les fonctions d'interpolation dans l'approximation de la variable inconnue ne sont plus fonction des coordonnées de l'élément réel, mais deviennent fonction des coordonnées de l'élément de référence. Il s'ensuit que pour tous les éléments réels liés à un même élément de référence, les fonctions d'interpolation garderont la même valeur $N(\epsilon)$, d'où une plus grande facilité dans les calculs.

IV

MODELE NUMERIQUE

(FORMULATION PAR ELEMENTS FINIS)

Nous avons déjà établi l'équation qui régit l'écoulement en milieu poreux saturé soit :

$$\text{Div}(\bar{k} \text{ grad } h) = C_s \frac{\partial h}{\partial t} + q \quad (5)$$

avec des conditions aux limites et des conditions initiales.

Nous allons passer de cette équation aux dérivées partielles à une formulation intégrale en utilisant la méthode des résidus pondérés.

4.1 FORMULATION INTEGRALE

Le résidu est défini par :

$$R(h) = -\text{div}(k \text{ grad } h) + C_s \frac{\partial h}{\partial t} + q$$

R est évidemment nul quand h est une solution exacte de l'équation (5)

La méthode des résidus pondérés consiste à chercher des fonctions ψ qui annulent la forme intégrale :

$$W(h) = \int_V \langle \psi \rangle \{ R(h) \} \quad \text{pour toutes}$$

fonctions de pondération ψ appartenant à

un ensemble de fonction $E\psi$, h appartenant à l'ensemble Eh des solutions admissibles qui satisfont les conditions aux limites et qui sont dérivables jusqu'à l'ordre m ($m=2$ dans notre cas).

Toutes solutions qui annulent le résidu annulent également la forme intégrale. Par contre la solution h de la forme intégrale dépend du choix des fonctions de pondération $E\psi$. En particulier si $E\psi$ est un ensemble fini alors une solution de la forme intégrale est une solution approximative du problème. Le résidu dans ce cas n'est plus nul. Dans notre étude nous choisirons un ensemble $E\psi$ fini.

Reprenons l'équation (5)

$$-\text{div} (\bar{k} \text{grad} h) + q + c_s \frac{\partial h}{\partial t} = 0$$

que nous notons encore

$$-\sum_j \left(k^{ij} \frac{\partial h}{\partial x_i} \right) + q + c_s \frac{\partial h}{\partial t} = 0$$

avec x_i coordonnée suivant l'axe i d'un point x du domaine.

La forme intégrale s'écrit :

$$w(h) = \int_V \gamma \left(-\frac{\partial}{\partial x_j} \left(k^{ij} \frac{\partial h}{\partial x_i} \right) \right) dv + \int_V \gamma \left(q + c_s \frac{\partial h}{\partial t} \right) dv = 0$$

Après intégration par parties, nous obtenons :

$$w(h) = \int_V \frac{\partial \gamma}{\partial x_j} k^{ij} \frac{\partial h}{\partial x_i} dv + \int_S \gamma \left(-k^{ij} \frac{\partial h}{\partial x_i} \right) \cdot n_j ds + \int_V \gamma \left(q + c_s \frac{\partial h}{\partial t} \right) dv = 0 \quad (6)$$

Le terme $q_s = \left(-k^{ij} \frac{\partial h}{\partial x_i} \right) \cdot n_j$ représente un flux à travers la limite s du domaine V . q_s est un flux surfacique quand V est un espace (3-D) et un flux linéaire quand V représente un espace (2-D)

L'intégration par parties diminue les conditions de dérivabilité sur la variable (h) et fait apparaître un flux (q_s). Par contre elle augmente les conditions de dérivabilité de (4)

4.2 DISCRETISATION DE LA FORME INTEGRALE

En faisant une approximation par éléments finis de la variable h , nous obtenons :

$$h_{app} = \hat{h}(x, H_n) = \langle N \rangle \{ H_n \}$$

En l'introduisant dans la forme intégrale, celle-ci devient :

$$w(\vec{h}) = \int_V \psi R(\vec{h}(x, H_n)) dv = 0$$

nous choisissons n fonctions de pondération indépendantes (ψ_1, \dots, ψ_n) et puisque l'équation $w(\vec{h}) = 0$ doit être vérifiée quelque soit le choix de ψ , nous aurons le système suivant :

$$w_1 = \int_V \psi_1 R(\vec{h}) dv = 0$$

$$w_2 = \int_V \psi_2 R(\vec{h}) dv = 0$$

$$\vdots$$

$$w_n = \int_V \psi_n R(\vec{h}) dv = 0$$

ce qui constitue un système d'équations algébriques (n équations à n inconnues).

Remarque : Nous avons procédé ici à une discrétisation de la variable h et non du domaine. l'élément est à ce stade confondu au domaine entier et $\{H_n\}$ représente le vecteur composé des valeurs de la variable à tous les nœuds d'interpolation.

4.3 CHOIX DES FONCTIONS DE PONDERATION

Suivant le choix de la fonction de pondération, on aboutit à différentes méthodes, méthodes dont les plus importantes sont :

- . Collocation par point
- . Collocation par sous-domaines
- . méthode des moindres carrés
- . méthode de Galerkin

Nous utiliserons celle de Galerkin. Dans cette méthode, l'ensemble \mathcal{E}_h est constitué par des variations S_h des fonctions inconnues h .

$$\text{soit } \mathcal{Y} = S_h = \langle N \rangle \{S_{hn}\}$$

où $\{S_{hn}\}$ représente un vecteur colonne des variations des paramètres d'approximation (variables nodales)

la relation $\langle N \rangle \{S_{hn}\}$ peut encore s'écrire $\langle S_{hn} \rangle \{N\}$. ce qui nous permet d'exprimer la forme intégrale sous la forme :

$$W(\vec{h}) = \langle S_{hn} \rangle \int_V \{N\} R(\vec{h}) dV = 0$$

Par la suite, nous noterons $\vec{h} \equiv h$.

Si dans l'équation $w(R) = 0$ nous faisons n choix de $\langle Sh_n \rangle$ du type $\langle 1, 0, \dots, 0_n \rangle$, nous aurons n équations suivantes :

$$w_1(R) = \int_V N_1 R(R) dV = 0$$

$$w_2(R) = \int_V N_2 R(R) dV = 0$$

⋮

$$w_n(R) = \int_V N_n R(R) dV = 0$$

que nous noterons

$$w(R) \equiv \begin{Bmatrix} w_1 \\ \vdots \\ w_n \end{Bmatrix} = \int_V \begin{Bmatrix} N_1 \\ \vdots \\ N_n \end{Bmatrix} R(R) dV = \{0\}$$

ou plus simplement

$$w(R) = \int_V \{N\} R(R) dV = 0 \quad (7)$$

4.4 DISCRETISATION DU DOMAINE ET FORMULATION MATRICIELLE

Le domaine d'intégration V est divisé en domaines élémentaires v^e et l'intégrale sur V devient une somme d'intégrales sur chaque élément.

nous aurons l'égalité :

$$W(h) = \int_V \{N\} R(h) dV = \sum_{e=1}^m \int_{V^e} \{N\}^e R(h)^e dV^e = 0$$

que nous notons :

$$W^e(h) = \sum_{e=1}^m W^e(h)$$

$W^e(h)$ représente la forme intégrale du résidu pondéré élémentaire.

Revenons à l'équation (6) que nous considérons en nous limitant au domaine élémentaire V^e et substituons h par son approximation par éléments finis soit $h = \langle N \rangle \{h_n\}^e$

(ici toutes les relations sont définies dans l'élément et ne font intervenir que les variables nodales de l'élément.)

nous aurons :

$$W^e(h) = \begin{Bmatrix} W_1^e \\ \vdots \\ W_n^e \end{Bmatrix} = \int_{V^e} \{N\}^e (\text{div} - k \text{ grad} \langle N \rangle \{h_n\}^e) dV \\ + \int_{V^e} \{N\}^e \left(q + c_s \frac{\partial h}{\partial t} \right) dV^e$$

ce qui peut encore s'écrire: (on intègre par parties)

$$\begin{aligned}
 W^e(h) = & \int_{V^e} \frac{\partial \{N\}^e}{\partial x_j} k_{ij} \frac{\partial \langle N \rangle^e}{\partial x_i} dV^e \{h_n\}^e \\
 & + \int_{S^e} \{N\}^e q_s ds^e + \int_{V^e} \{N\}^e q dV^e \\
 & + \int_V c_s \{N\}^e \langle N \rangle^e \frac{\partial \{h_n\}^e}{\partial t} dV^e \quad (8)
 \end{aligned}$$

avec

$$\frac{\partial \{N\}^e}{\partial x_j} = \begin{Bmatrix} \frac{\partial N_1}{\partial x_j} \\ \vdots \\ \frac{\partial N_n}{\partial x_j} \end{Bmatrix} \quad \text{et} \quad \frac{\partial \langle N \rangle^e}{\partial x_i} = \left\langle \frac{\partial N_1}{\partial x_i} \quad \dots \quad \frac{\partial N_n}{\partial x_i} \right\rangle$$

si nous notons

$$[K] = \int_{V^e} \frac{\partial \{N\}^e}{\partial x_j} k_{ij} \frac{\partial \langle N \rangle^e}{\partial x_i} dV^e$$

$$\{F\} = q_s \int_{S^e} \{N\}^e ds^e + \int_{V^e} \{N\}^e q dV^e$$

$$[m] = c_s \int_{V^e} \{N\}^e \langle N \rangle^e dV^e$$

alors la forme intégrale élémentaire peut s'écrire

$$W^e(h) = [K] \{h_n\}^e + \{F\} + [m] \frac{\partial \{h_n\}^e}{\partial t}$$

$[K]$ est appelé matrice de rigidité élémentaire

$\{f\}$ est le vecteur élémentaire de sollicitation

$[m]$ est la matrice masse élémentaire

La forme intégrale globale s'écrit:

$$W(R) = \sum_{e=1}^m [k] \{h_n\} + \{f\} + [m] \frac{\partial \{h_n\}}{\partial t} = 0$$

$$\text{soit } W(R) = [K] \{h_n\} + \{F\} + [M] \frac{\partial \{h_n\}}{\partial t} = 0$$

Pour une présentation plus élégante de l'équation (8), nous allons noter

$$B = \begin{bmatrix} \frac{\partial \langle N \rangle^e}{\partial x_1} \\ \vdots \\ \frac{\partial \langle N \rangle^e}{\partial x_n} \end{bmatrix}$$

ce qui donnera

$$W(R) = \int_{V_e} B^T \bar{k} B dV^e \{h_n\} + \int_{S_e} q_s \{N\}^e dS_e \\ + \int_V \{N\}^e q_v dV^e + \int_{V_e} c_s \{N\}^e \frac{\partial \{h_n\}}{\partial t} dV^e$$

4.5 INTRODUCTION DES ELEMENTS DE REFERENCE

En établissant une relation géométrique entre l'espace de référence et l'espace réel tel que:

$$x(\epsilon) = [\bar{N}(\epsilon) \rangle \{x_n\}$$

nous aurons

$$h(x) = \langle N(x) \rangle \{h_n\} = \langle N(\epsilon) \rangle \{h_n\}$$

nous choisissons des éléments isoparamétriques ce qui nous donne $\bar{N} \equiv N$.

En exprimant la matrice $[B]$ en fonction des coordonnées de l'élément de référence nous obtenons

$$[B] = \begin{bmatrix} \frac{\partial \langle N \rangle}{\partial x_1} \\ \frac{\partial \langle N \rangle}{\partial x_2} \\ \vdots \\ \frac{\partial \langle N \rangle}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial \langle N \rangle}{\partial \eta_1} \frac{\partial \eta_1}{\partial x_1} + \frac{\partial \langle N \rangle}{\partial \eta_2} \frac{\partial \eta_2}{\partial x_1} + \dots + \frac{\partial \langle N \rangle}{\partial \eta_n} \frac{\partial \eta_n}{\partial x_1} \\ \frac{\partial \langle N \rangle}{\partial \eta_1} \frac{\partial \eta_1}{\partial x_2} + \frac{\partial \langle N \rangle}{\partial \eta_2} \frac{\partial \eta_2}{\partial x_2} + \dots + \frac{\partial \langle N \rangle}{\partial \eta_n} \frac{\partial \eta_n}{\partial x_2} \\ \vdots \\ \frac{\partial \langle N \rangle}{\partial \eta_1} \frac{\partial \eta_1}{\partial x_n} + \frac{\partial \langle N \rangle}{\partial \eta_2} \frac{\partial \eta_2}{\partial x_n} + \dots + \frac{\partial \langle N \rangle}{\partial \eta_n} \frac{\partial \eta_n}{\partial x_n} \end{bmatrix}$$

ce qui peut s'écrire

$$[B] = \begin{bmatrix} \frac{\partial \eta_1}{\partial x_1} & \frac{\partial \eta_2}{\partial x_1} & \dots & \frac{\partial \eta_n}{\partial x_1} \\ \frac{\partial \eta_1}{\partial x_2} & \frac{\partial \eta_2}{\partial x_2} & \dots & \frac{\partial \eta_n}{\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \eta_1}{\partial x_n} & \frac{\partial \eta_2}{\partial x_n} & \dots & \frac{\partial \eta_n}{\partial x_n} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial \langle N \rangle}{\partial \eta_1} \\ \frac{\partial \langle N \rangle}{\partial \eta_2} \\ \vdots \\ \frac{\partial \langle N \rangle}{\partial \eta_n} \end{bmatrix} = [Q] \cdot [B]$$

$[Q]$ est l'inverse de la matrice jacobienne $[J]$.

la matrice de rigidité $[K]$ s'écrit

$$[K] = \int_{V^e} [B_\xi]^T \cdot [Q]^T \cdot \bar{k} \cdot [Q] \cdot [B_\xi] dV^e.$$

avec le changement de variable on a :

$$dV = \det(J) dy_1 dy_2 dy_3 = \det(J) d\xi$$

la matrice de rigidité devient :

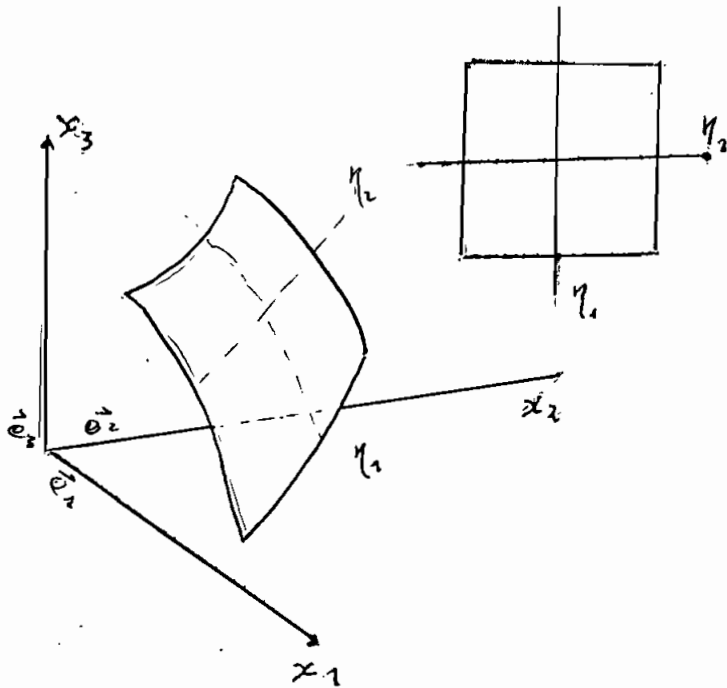
$$[K] = \int_{\xi} [B_\xi]^T \cdot [Q]^T \cdot \bar{k} \cdot [Q] \cdot [B_\xi] \det(J) d\xi$$

4.6. MODELE 1-D, 2-D dans un espace 3-D

L'équation ci-dessus est valable quand on travaille dans un espace $n-D$ avec des éléments $n-D$. Mais afin de rendre notre logiciel assez flexible, nous avons utilisé des éléments 1-D et 2-D dans un espace 3-D. Dans ces conditions, nous obtenons une matrice jacobienne non inversible ce qui rend impossible l'utilisation de la formule ci-dessus. Pour contourner ce problème, nous proposons une méthode générale basée sur la représentation des gradients en coordonnées curvilignes.

(KIRALY, 1979, KLINGBELL 1966, TEICHMANN 1964)

En espace 2-D nous avons la situation suivante



ce qui donne les relations suivantes :

$$\vec{r}(\eta_1, \eta_2) = x_1 \vec{e}_1 + x_2 \vec{e}_2 + x_3 \vec{e}_3$$

$$x_i(\eta_1, \eta_2) = \langle N \rangle \{x_i\}$$

en espace 2-D on a la base covariante définie par $\vec{a}_k = \frac{\partial \vec{r}}{\partial \eta_k}$

le tenseur métrique covariant est noté

$$g^{ik} = [\vec{a}_i \cdot \vec{a}_k]$$

si on désigne par \$\psi^n\$ les valeurs nodales nous avons l'approximation

$$\psi(\eta_1, \eta_2) = \langle N \rangle \{\psi^n\}$$

ce qui donne la relation

$$\text{grad } \varphi = \frac{\partial \vec{r}}{\partial \eta_k} g_{ik} \frac{\partial \varphi}{\partial \eta_i} = \frac{\partial \vec{r}}{\partial \eta_k} g_{ik} \frac{\partial \langle N \rangle}{\partial \eta_i} \{P^n\}$$

avec $g_{ik} = g^{ik^{-1}}$ vecteur contravariant

de façon plus explicite et en substituant h à φ nous obtenons

$$\text{grad } h = \vec{e}_j \left[\frac{\partial x_j}{\partial \eta_k} g_{ik} \frac{\partial \langle N \rangle}{\partial \eta_i} \right] \{h_n\}$$

$\{h_n\}$ étant le vecteur colonne des valeurs nodales

A ce stade il est facile d'en déduire la matrice des gradients qui s'écrit :

$$B^j = \frac{\partial x_j}{\partial \eta_k} g_{ik} \frac{\partial \langle N \rangle}{\partial \eta_i}$$

B^j étant la colonne j de la matrice B

avec cette transformation on a, pour les éléments

$$1-D \quad dV = (\det g_{ik})^{1/2} d\eta$$

$$2-D \quad dV = (\det g_{ik})^{1/2} d\eta_1 d\eta_2$$

$$3-D \quad dV = (\det g_{ik})^{1/2} d\eta_1 d\eta_2 d\eta_3$$

4.7) METHODE D'INTEGRATION NUMERIQUE

Il n'est pas toujours possible d'avoir (vue la complexité des équations) une intégration directe des équations. On a donc recours à des méthodes numériques d'intégration.

Les méthodes consistent d'une façon générale à utiliser des points d'intégrations x et des coefficients de pondération w tel que l'on puisse avoir :

$$\int f(x) dx = \sum_{i=1}^p w_i f(x_i)$$

Pour $f(x)$ polynomiale, un choix suffisant du nombre de points d'intégration peut conduire à des solutions exactes. Mais quand on traite les éléments compliqués (de forme), on obtient des intégrales de fonctions rationnelles dont l'intégration numérique ne donne qu'une valeur approchée. Dans tous les cas, un choix judicieux des points d'intégration et de leur nombre, donnera une précision satisfaisante.

Le choix des points d'intégration et des coefficients de pondération conduit à différentes méthodes :

- . Méthode de Gauss
- . Méthode de Newton-Cotes
- . Méthode directe

Nous allons présenter la méthode utilisée qui est celle de Gauss.

4.7.1 Intégration à une dimension :

On détermine n coefficients w_i et n points d'intégrations ξ_i de manière à intégrer exactement des polynômes d'ordre $m < 2n-1$.
Pour un élément de référence cela revient à écrire

$$\int_{-1}^1 \gamma(\xi) d\xi = \sum_{i=1}^n w_i \gamma(\xi_i)$$

avec $\gamma(\xi) = a_1 + a_2 \xi + \dots + a_{2n-1} \xi^{2n-1}$

Nous utiliserons trois points de Gauss, ce qui donnera des résultats exactes si la fonction à intégrer est un polynôme de degré inférieur ou égal à $(2n-1)$ soit 5

4.7.2 Intégration à deux dimensions :

Elle consiste à utiliser une intégration numérique à une dimension dans chaque direction. Si on utilise r_1 points dans le sens η_1 et r_2 points dans le sens η_2 on intègre exactement le produit d'un polynôme en η_1 d'ordre $2r_1 - 1$ et d'un polynôme en η_2 d'ordre $2r_2 - 1$. Elle utilise donc $r = r_1 \cdot r_2$ points d'intégration.

Pour les éléments de référence carré, nous utilisons neuf points d'intégration ce qui donne pour une fonction $\gamma(\eta_1, \eta_2)$ à intégrer :

$$\int_{-1}^{+1} \int_{-1}^{+1} \gamma(\eta_1, \eta_2) d\eta_1 d\eta_2 = \sum_{i=1}^3 \sum_{j=1}^3 w_i w_j \gamma(\eta_{1i}, \eta_{2j}).$$

Pour les éléments triangulaires, nous avons utilisé la méthode (de Gauss) avec sept points d'intégration.

soit

$$\int_0^1 \int_0^{1-\eta_1} \gamma(\eta_1, \eta_2) d\eta_1 d\eta_2 = \sum_{i=1}^7 w_i \gamma(\eta_{1i}, \eta_{2i})$$

4.3.3 Intégration à trois dimensions

Pour les éléments de référence cubiques on procède à une intégration numérique à une dimension dans chaque direction. ce qui donne :

$$\int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \gamma(\eta_1, \eta_2, \eta_3) d\eta_1 d\eta_2 d\eta_3 = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} w_i w_j w_k \gamma(\eta_{i,j,k})$$

Une intégration par la méthode directe est utilisée pour les éléments tétraédriques.

Nous présenterons ultérieurement une étude sur la précision de l'intégration numérique. (se référer à l'annexe pour le choix des points de Gauss et les coefficients de pondération).

V

ASSEMBLAGE ET RESOLUTION

5. ASSEMBLAGE ET RESOLUTION

L'intégration numérique nous permet d'attribuer des valeurs numériques à toutes les composantes de la matrice élémentaire définie précédemment. A la suite de la sommation de toutes les matrices élémentaires composant le domaine, on obtient une matrice globale. Cette opération, appelée assemblage, est évidemment étendue aux autres membres de l'équation. Au stade nous obtenons un système global d'équation qu'il faut résoudre. La taille de la matrice globale varie dans le même sens que le nombre d'éléments. Elle contient en général un nombre considérable de "zéro". Le traitement numérique d'une telle matrice entraîne des opérations inutiles avec des "zéro" et demande une sollicitation intensive de la mémoire vive (RAM). En vue de palier à ce problème et d'optimiser le traitement, des techniques d'assemblage et de résolution ont été développées.

Nous utilisons la méthode frontale qui consiste à éliminer, après assemblage et élimination de Gauss, les variables correspondantes à des points qui n'interviendront plus. Les lignes ainsi éliminées seront occupées par la suite par des variables (ou lignes) qui apparaissent pour la première fois lors d'assemblages futurs. On évite ainsi de "travailler avec des zéros" et on réduit considérablement la taille de la matrice globale résidant en mémoire. L'inconvénient majeur de cette méthode est qu'elle doit être combinée à la méthode d'élimination de Gauss, alors que pour la résolution des systèmes d'équation on lui préfère des méthodes itératives.

5.1 Présentation de la méthode Frontale

Soit une matrice G de dimension $N \times N$ qui admet une élimination de Gauss. Elle sera donc transformée en une matrice supérieure.

Considérons la $n^{\text{ième}}$ élimination de Gauss.

Nous avons la matrice suivante :

$$\begin{array}{c} r \\ i \end{array} \begin{array}{c} r \quad j \\ \left[\begin{array}{cc} g_{rr} & g_{rj} \\ g_{ir} & g_{ij} \end{array} \right] \end{array}$$

Nous aurons g_{ij} reçoit $g_{ij} - \frac{g_{ir}}{g_{rr}} \cdot g_{rj}$
Pour effectuer cette opération il est nécessaire de connaître uniquement g_{rr} , g_{rj} et g_{ir} . Il n'est donc pas nécessaire de connaître la valeur initiale de g_{ij} . On peut donc amorcer la triangulation avant d'avoir assemblé tous les éléments.

Voyons comment ce concept est appliqué dans la méthode frontale.

Considérons un domaine V subdivisé en k_p éléments. Chaque élément v_i ayant n_i nœuds.

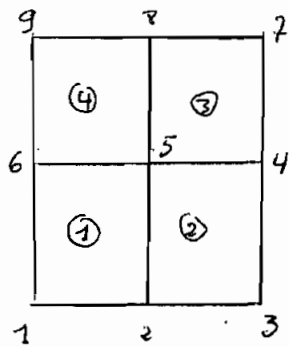
Pour chaque élément, on calcule une matrice élémentaire G_{EK} qui sera assemblée

dans une matrice de travail B . Ensuite on identifie les variables qui apparaissent pour la dernière fois. On procédera alors à l'élimination de Gauss sur cette variable, puis on élimine la ligne et la colonne correspondantes.

En effet dans la méthode d'assemblage, on vérifie que, quand toutes les matrices élémentaires qui ont en commun un nœud sont assemblées, la ligne et la colonne correspondantes à cette variable ne recevra plus de contribution dans la suite de l'assemblage. Nous constatons donc qu'il est possible de procéder à une élimination de Gauss en considérant comme pivot g_{rr} (r étant la ligne relative à la variable à éliminer), puisque à ce stade toutes les valeurs qui interviennent dans l'opération sont toutes connues (g_{rr} , g_{rj} , g_{ir}).

Pour illustrer la méthode, nous allons traiter un exemple complet de problème.

5.2 Exemple de calcul



Un domaine V composé de 4 éléments et 4 nœuds par élément, a un total de 9 nœuds.

En vue de reconnaître l'état d'apparition d'une variable nous utilisons des codes:

code	signification
n	$n^{\text{ième}}$ de n apparitions
-1	apparition intermédiaire
1	première et dernière apparition
0	Dernière apparition

A chaque dernière apparition, on procédera à une élimination de bords si nécessaire.

Pour ce problème, nous prendrons toutes les matrices élémentaires 6×6 identiques

à

$$\begin{bmatrix} 4 & 1 & 1 & 1 \\ 1 & 4 & 1 & 1 \\ 1 & 1 & 4 & 1 \\ 1 & 1 & 1 & 4 \end{bmatrix}$$

et tous les vecteurs de sollicitation identiques à $F_{EK} = \langle 4, 4, 4, 4 \rangle$.

Les vecteurs V_{EK} sont composés des numéros des variables. Les vecteurs C_{EK} contiennent les codes d'apparition des variables. La matrice de travail sera notée G_{EG} et le vecteur global de sollicitation F_{EG} .

1^{er} élément

Les matrices G_{EK} et G_{EG} sont identiques et on a :

V_{EG}	C_{EK}	G_{EG}	F_{EG}
$\begin{bmatrix} 1 \\ 2 \\ 5 \\ 6 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 2 \\ 4 \\ 2 \end{bmatrix}$	$\begin{bmatrix} 4 & 1 & 1 & 1 \\ 1 & 4 & 1 & 1 \\ 1 & 1 & 4 & 1 \\ 1 & 1 & 1 & 4 \end{bmatrix}$	$\begin{bmatrix} 4 \\ 4 \\ 4 \\ 4 \end{bmatrix}$

nous avons $C_{EK}[i] = 1$ ce qui signifie que

la variable n° 1 doit être éliminée, après élimination de Gauss. Les informations qui seront stockées sont:

Le numéro du noeud, les coefficients, et la dimension de la matrice de travail.

soit $\{1 / 4 \quad 1 \quad 1 \quad 1 / 4 \quad 1 / 4\}$

Après élimination nous obtenons:

VEG	CEK	GEG	FEG
$\begin{bmatrix} - \\ - \\ 2 \\ 5 \\ 6 \end{bmatrix}$	$\begin{bmatrix} - \\ 2 \\ 4 \\ 2 \end{bmatrix}$	$\begin{bmatrix} - & - & - & - \\ 1 & 3,75 & 0,75 & 0,75 \\ - & 0,75 & 3,75 & 0,75 \\ - & 0,75 & 0,75 & 3,75 \end{bmatrix}$	$\begin{bmatrix} - \\ 3 \\ 3 \\ 3 \end{bmatrix}$

Assemblage de l'élément n° 2. Pour cet élément on a $VEK < 2, 3, 4, 5 >$

$CEK < 0, 1, 2, -1 >$

la matrice globale devient

$$\begin{array}{c}
 \left[\begin{array}{c} 3 \\ 2 \\ 5 \\ 6 \\ 4 \end{array} \right] \quad \left[\begin{array}{cccccc} 4 & 1 & 1 & 0 & -1 \\ 1 & 7,75 & 1,75 & 0,75 & 1 \\ 1 & -1,75 & 7,75 & 0,75 & 1 \\ 0 & 0,75 & 0,75 & 3,75 & 0 \\ 1 & 1 & 1 & 0 & 4 \end{array} \right] \quad \left[\begin{array}{c} 4 \\ 7 \\ 7 \\ 3 \\ 4 \end{array} \right]
 \end{array}$$

On note l'apparition des variables N°3 et N°4. La variable N°3 a pris la place que la variable (1) avait occupée dans matrice de travail. Une nouvelle place a été créée pour la variable (4). Le vecteur CEX montre qu'il faut éliminer les variables (3) et (2).

Pour la variable (3) nous stockons:

$$|3| \quad 4 \quad 1 \quad 1 \quad 0 \quad -1 \quad | \quad 4 \quad | \quad 5 \quad |$$

Pour la variable (2) on a:

$$|2| \quad 7,5 \quad , \quad 1,5 \quad , \quad 0,75 \quad , \quad 0,75 \quad | \quad 6 \quad | \quad 4 \quad |$$

La matrice de travail devient:

$$\begin{array}{c}
 \left[\begin{array}{c} - \\ - \\ 5 \\ 6 \\ 4 \end{array} \right] \quad \left[\begin{array}{cccccc} - & - & - & - & - \\ - & - & - & - & - \\ - & - & 7,2 & 0,6 & - & 0,6 \\ - & - & 0,6 & 3,575 & - & 0,075 \\ - & - & 0,6 & - & 0,075 & 3,675 \end{array} \right] \quad \left[\begin{array}{c} - \\ - \\ 4,8 \\ 2,4 \\ 2,4 \end{array} \right]
 \end{array}$$

Assemblage de l'élément (3) on a

$$V_{EK} < 5, 4, 7, 8 >$$

$$C_{EK} < -1, 0, 1, 2 >$$

la matrice globale devient :

$$\begin{bmatrix} 7 \\ 8 \\ 5 \\ 6 \\ 4 \end{bmatrix} \begin{bmatrix} 4 & 1 & -1 & 0 & 1 \\ 1 & 4 & 1 & 0 & 1 \\ 1 & 1 & 11,2 & 0,6 & 1,6 \\ 0 & 0 & 0,6 & 3,675 & -0,075 \\ 1 & 1 & 1,6 & -0,075 & 7,675 \end{bmatrix} \begin{bmatrix} 4 \\ 4 \\ 9,8 \\ 2,4 \\ 6,4 \end{bmatrix}$$

l'élimination de la variable (2) donne :

$$|7| 4, 1, 1, 0, 1 | 4 | 5 |$$

Pour la variable (4) on a :

$$|4| 0,75, 1,35, -0,075, 7,425 | 6,15 | 4 |$$

La matrice globale devient :

$$\begin{bmatrix} - \\ 8 \\ 5 \\ 6 \\ - \end{bmatrix} \begin{bmatrix} - & - & - & - & - \\ - & 3,6742 & 0,6136 & 0,0075 & - \\ - & 0,6136 & 10,204 & 0,6136 & - \\ - & 0,0075 & 0,6136 & 3,675 & - \\ - & - & - & - & - \end{bmatrix} \begin{bmatrix} - \\ 2,45 \\ 7,81 \\ 2,45 \\ - \end{bmatrix}$$

Assemblage de l'élément (4) : on a

$$VEK < 6 \quad 5 \quad 8 \quad 9 >$$

$$LEK < 0 \quad 0 \quad 0 \quad 1 >$$

la matrice globale s'écrit :

$$\begin{bmatrix} 9 \\ 8 \\ 5 \\ 6 \end{bmatrix} \begin{bmatrix} 4 & 1 & 1 & 1 \\ 1 & 7,6742 & 1,6136 & 1,0075 \\ 1 & 1,6136 & 14,204 & 1,6136 \\ 1 & 1,0075 & 1,6136 & 7,67 \end{bmatrix} \begin{bmatrix} 4 \\ 6,45 \\ 11,81 \\ 6,45 \end{bmatrix}$$

L'élimination des variables 6, 5, 8, 9 nous donne respectivement les équations suivantes :

$$|6| \quad 1 \quad 1,075 \quad 1,6136 \quad 7,67 \quad | \quad 6,45 \quad | \quad 4 |$$

$$|5| \quad 0,789 \quad 1,4 \quad 13,86 \quad | \quad 9,7 \quad | \quad 3 |$$

$$|8| \quad 0,781 \quad 7,4 \quad | \quad 4,55 \quad | \quad 2 |$$

$$|9| \quad 3,74 \quad | \quad 2,1 \quad | \quad 1 |$$

Nous allons passer à la substitution inverse pour le calcul des inconnues.

Substitution inverse :

Pour calculer les variables, nous faisons appel aux équations en sens inverse en commençant par la dernière équation stockée. Pour le calcul d'une variable, nous avons besoin de connaître les variables qui composaient la matrice au moment de son élimination. Ce qui donnera pour notre cas :

$$h_9 \times 3,74 = 2,1 \quad \Rightarrow \quad h_9 = 0,56$$

Pour la variable 8 nous aurons

$$h_8 \times 7,4 + h_9 \times 0,788 = 4,55$$

$$\text{soit } h_8 = 0,56$$

Par cette même démarche nous obtenons

$$h_5 = 0,57$$

$$h_6 = 0,58$$

$$h_4 = 0,57$$

$$h_2 = 0,57$$

$$h_7 = 0,57$$

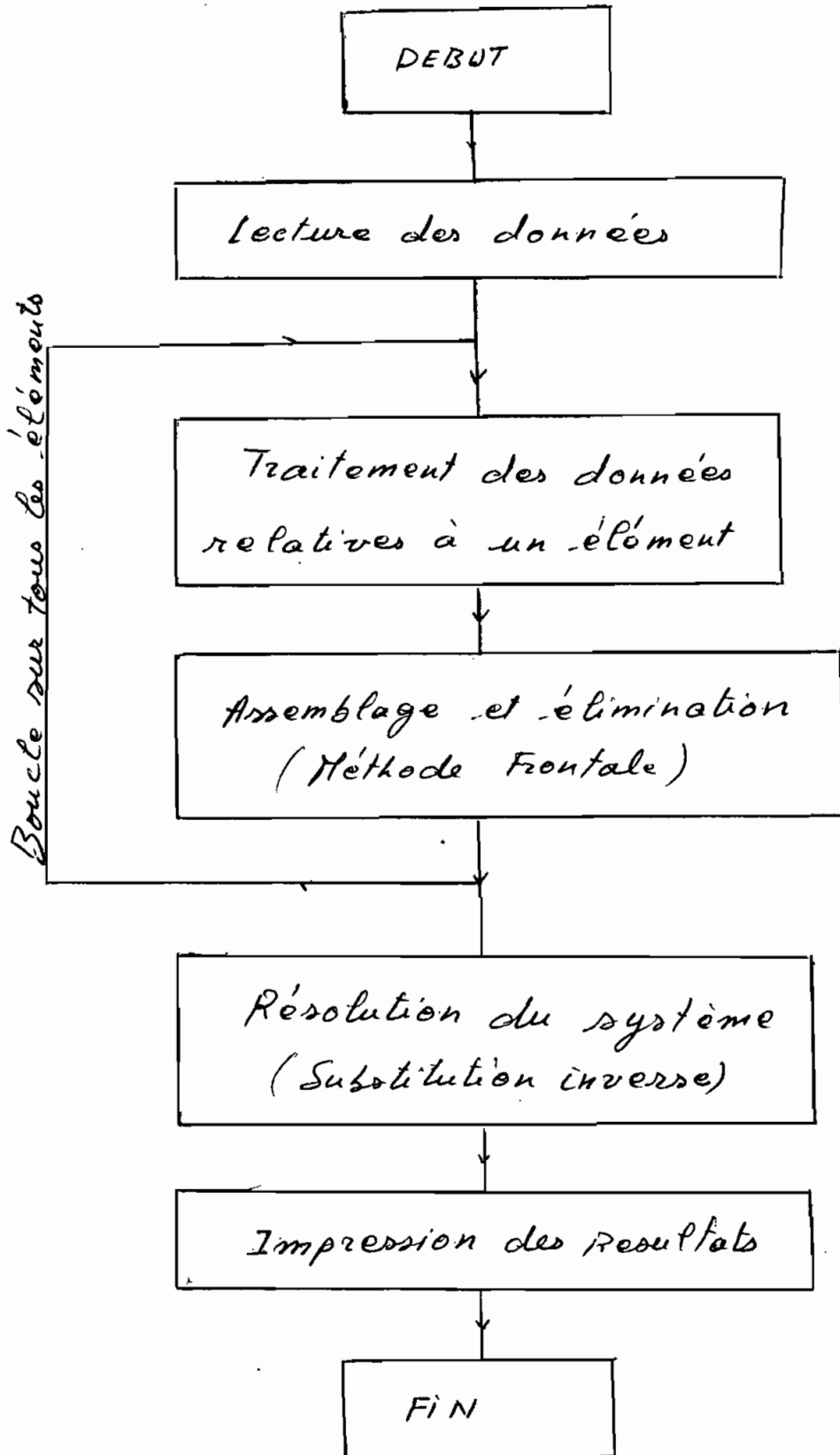
$$h_3 = 0,56$$

$$h_1 = 0,57$$

VI

PRESENTATION DU LOGICIEL

6.1 ORGANIGRAMME DU PROGRAMME



6.2 LECTURE DES DONNEES

Nous disposons de trois fichiers pour la lecture des données.

1 - Un fichier de nom physique NENOPERM.DAT et de nom logique Fiche_alpha qui enregistre les tenseurs de perméabilité. Le domaine doit être divisé en classe de perméabilité, les éléments de même classe ont la même perméabilité.

L'enregistrement comporte :

- un numéro qui indique la classe et six coefficients qui représentent les valeurs de la matrice supérieure du tenseur.

2 - Un fichier de nom physique NENO.DAT et de nom logique local, constitue la table de connectivité. Il enregistre :

• le type d'élément qui peut être

36 : Pour les éléments triangulaires à 6 nœuds.

48 : Pour les éléments quadrilatéraux à 8 nœuds

49 : Pour éléments quadrilatéraux à 9 nœuds.

23 : Pour les éléments linéaires à 3 nœuds.

- la classe qui indique la classe de perméabilité à laquelle appartient l'élément
- l'alimentation distribuée ALD
- les numéros des nœuds de l'élément.

3. Enfin un dernier fichier de nom physique NEND.LOR et de nom logique CORG. Son enregistrement comprend :

- les coordonnées de chaque nœud
- Un facteur -1 ou 0 qui indique que la charge au nœud est connue (-1) ou pas (0).

Pour une charge connue, nous enregistrons sa valeur dans la variable (VALK). Enfin nous enregistrons une dernière valeur dans la variable VALDSP qui représente un débit spécifique au nœud, positif en cas d'injection et négatif pour un prélèvement.

Pour l'exécution, nous disposons de deux options :

- Modifier les données existantes
- Créer de nouvelles données.

Dans ce cas le fichier est réouvert et les anciennes valeurs seront effacées.

Les données et les résultats sont en unité SI (système International).

6.3 TRAITEMENT D'UN ELEMENT

Disposant de toutes les données relatives à l'élément, nous pouvons passer au traitement de cet élément.

Pour chaque point d'intégration nous évaluons la matrice élémentaire k_e et le vecteur élémentaire F_{EK} . Nous formons une boucle sur tous les points de Gauss. A l'issue de cette opération nous avons la matrice élémentaire G_{EK} et le vecteur élémentaire F_{EK} qui seront assemblés dans la matrice globale G_{EG} et F_{EG} le vecteur global de sollicitation F_{EG} . C'est alors que nous faisons un test sur le vecteur C_{EK} qui nous donne les codes d'apparition des nœuds.

Pour une réponse positive à ce test, nous ferons un deuxième test sur la charge. Si celle-ci est inconnue nous procéderons à une élimination de Gauss standard dans le cas contraire faisons une élimination non standard²².

L'équation de la variable éliminée est stockée dans un fichier sous le nom "équax" et sa place dans la matrice globale est libérée par une mise à zéro.

À l'issue de ce traitement qui sera fait pour tous les éléments, nous passons à la résolution (calcul des inconnues). Nous précisons que les inconnues peuvent être des charges ou des débits.

²² soit un système

$$\begin{cases} a_1x + b_1y + c_1z = e_1 \\ a_2x + b_2y + c_2z = e_2 \\ a_3x + b_3y + c_3z = e_3 \end{cases}$$

si la valeur de x est connue l'élimination de Gauss non standard donnera :

$$a_1x_0 + b_1y + c_1z = e_1$$

$$b_2y + c_2z = e_2 - a_2x_0$$

$$b_3y + c_3z = e_3 - a_3x_0$$

6.4 PRECISION DES CALCULS :

Les sources d'erreurs sont nombreuses. Mais elles peuvent être négligeable dans certaines conditions.

6.4.1 Les principales sources d'erreurs :

a - Idealisation du domaine d'étude :

Il n'est pas toujours possible dans la discrétisation d'épouser exactement le domaine global. On peut minimiser cette erreur en prenant des éléments de taille réduite.

b - Erreur d'intégration :

Elle augmente avec la distorsion de l'élément. Pour une intégration sur un segment ($r=3$), on estime l'erreur à $0,7 \times 10^{-2} \frac{d^{4y}}{dy^4}$. D'une façon générale, l'erreur est estimée à

$$e = \frac{2^{2r+1} (r!)^4}{(2r+2) [(2r)!]^3} \frac{d^{2r}}{dy^{2r}}$$

Pour un nombre de points d'intégration suffisant cette erreur devient très négligeable.

ϵ : erreur d'approximation:

On l'exprime en fonction de la dimension (P) de l'élément sous la forme

$$\epsilon \leq c P^{-\alpha}$$
 où c et α dépendent du type d'élément. Elle tend donc vers zéro avec la dimension de l'élément.

Il ressort à l'issue de ces observations que pour avoir le maximum de précision, il faut:

- utiliser des éléments de forme linéaire

- réduire la taille des éléments

- faire un choix judicieux des points de cours et de leur nombre.

Le faisant, on accroît le nombre d'opérations numériques nécessaires donc du temps d'exécution. Pour une calculatrice pouvant donner des précisions de l'ordre de (10^{-26}) , les erreurs dû aux traitements numériques sont négligeables.

(nous invitons le lecteur à se référer à l'annexe des résultats des tests enfin d'apprécier l'influence globale des erreurs)

6.5 TESTS DE VÉRIFICATION :

Nous invitons le lecteur à se référer à l'annexe pour les résultats des tests. Les tests de vérification ont porté sur les cas suivants :

- 1 - un élément du type 48
- 2 - un élément du type 49
- 3 - 4 éléments du type 48
- 4 - un cas concret d'écoulement vers un puits

ceci avec des conditions aux limites variables d'un cas à un autre ou au sein d'un même cas.

Pour le cas (3) nous avons obtenu des solutions identiques à ceux du Programme "FEM 200" (confère référence 8)

Pour le cas (4) nous avons utilisé l'équation de Dupuit pour chercher une solution analytique. le pourcentage d'erreur est de l'ordre de 0,02% pour les charges. Quant aux débits l'erreur y est négligeable.

CONCLUSION

Afin de rendre le logiciel assez complet et de permettre le traitement des cas concrets d'une grande dimension, il est vivement souhaité que ce travail soit poursuivi en une seconde phase. Les études de cette seconde phase comporteront entre autres,

- La résolution des problèmes d'écoulement transitoire.
- L'optimisation du temps de traitement.
- L'introduction des fonctions d'interpolation des éléments 3-0.

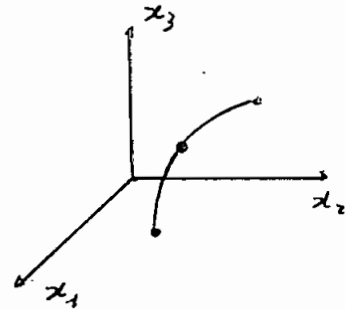
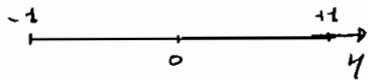
Les procédures pouvant traiter des éléments 3-0 sont disponibles dans le logiciel

REFERENCES

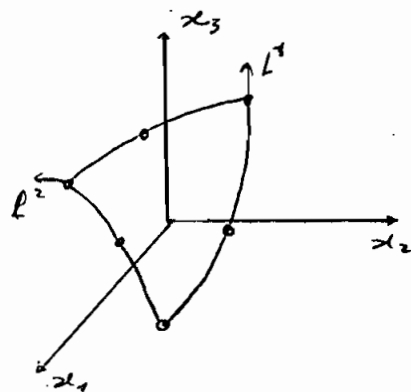
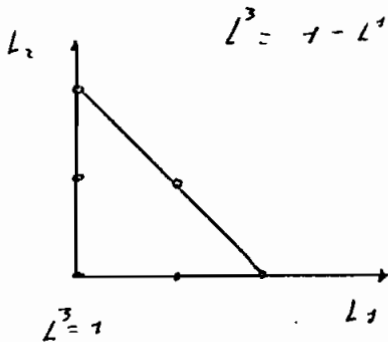
Eléments de référence Eléments réels en espace

3-0'

Segment à 3 nœuds



Elément triangulaire à 6 nœuds
(coordonnées barycentriques)



Elément rectangulaire à 8 ou 9 nœuds

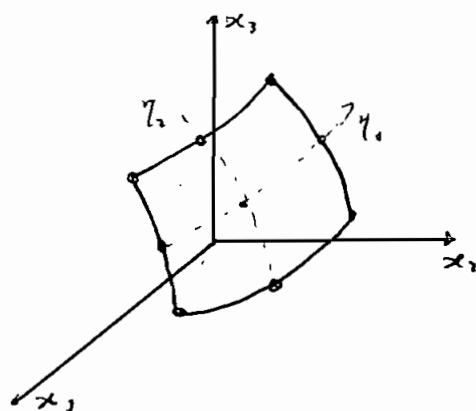
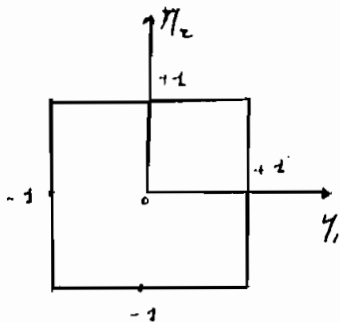
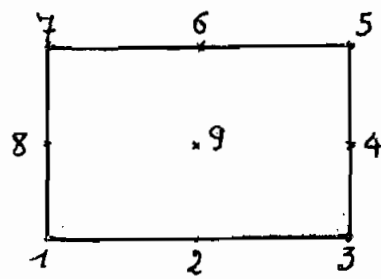
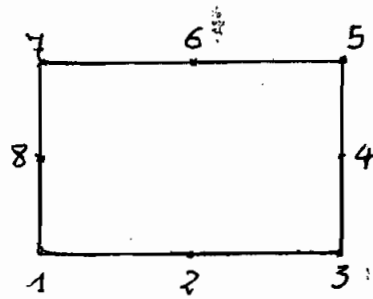
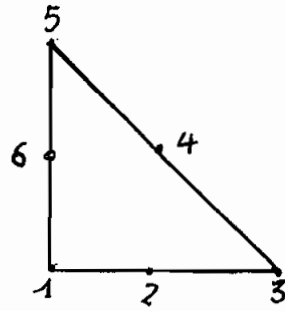
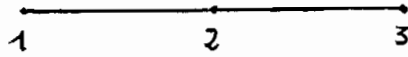


FIGURE DES ELEMENTS UTILISES



ORDRE DES NOEUDS DANS L'ELEMENT

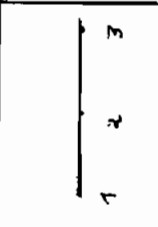
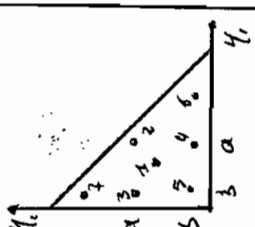
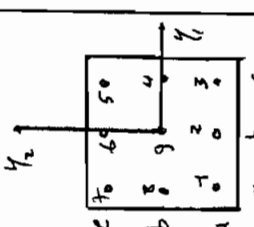
Élément	Point d'intégration r	COORDONNÉES y_1	COEFFICIENT DE pondération w
	3	$-\sqrt{3/5}$ $+0$ $+\sqrt{3/5}$	$5/9$ $8/9$ $5/9$
	4 $a = \frac{6 + \sqrt{15}}{21}$ $b = \frac{4}{7} - a$	$1/3$ a $1-2a$ a b $1-2b$ b $1-2b$	$9/40$ $A = \frac{155 + \sqrt{15}}{2400}$ $\frac{31}{240} - A$
	9 $c = \sqrt{3/5}$ $w_1 = 5/9$ $b = 0$ $w_2 = 8/9$ $a = -\sqrt{3/5}$ $w_3 = 5/9$	a b c c c b a a b	$w_1 \times w_1$ $w_1 \times w_2$ $w_1 \times w_3$ $w_2 \times w_3$ $w_3 \times w_3$ $w_2 \times w_2$ $w_1 \times w_3$ $w_1 \times w_2$ $w_2 \times w_2$

TABLEAU DES ÉLÉMENTS ET DES POINTS D'INTEGRATION

NOMENCLATURE

- FN : fonctions d'interpolation
- $FNDS, FNDU, FNDD$: dérivées des fonctions d'interpolation par rapport aux variables (η_1, η_2, η_3) de l'espace de référence
- $DIMELT$: Dimension d'un élément
- $NBND$: Nombre de nœuds d'un élément
- $NBTND$: Nombre total de nœuds.
- $NBELT$: Nombre total d'élément.
- IDI : Vecteur qui contient le nombre d'éléments auxquels est lié un nœud.
- IDR : Vecteur qui est identique à IDI au début de l'exécution, mais qui diminue de valeur au fur et à mesure qu'apparaît le nœud.
- CEK : Vecteur contenant les codes d'apparition.
- JS : Matrice composée des éléments de la base covariante.
- G : Tenseur métrique covariant
- ING : Tenseur métrique contravariant.

- W_S, W_T : les coefficients de pondération.
 ss, tk : les coordonnées des points d'intégration (2-0)
 $AL36, BL36, CL36$: les coordonnées barycentriques utilisées pour les éléments triangulaires.
 $PERM$: tenseur des perméabilités.
 XN, YN, ZN : les coordonnées des éléments réels.
 ke : Matrice de rigidité élémentaire calculée pour un point d'intégration.
 BEL : Matrice des gradients
 qtm : Vecteur qui représente pour les éléments 1-0, 2-0 et 3-0 respectivement des segments, des surfaces, des volumes d'influence.
 FEK : Vecteur de sollicitation élémentaire
 BIB : Matrice élémentaire de rigidité (somme de ke sur tous les points d'intégrations)
 GEG : Matrice de travail
 FEG : Vecteur de sollicitation de travail

- HHCXX : Vecteur contenant le chiffre (-1) ou 0 indiquant dans le 1er cas que la charge est connue et non connue dans le second.
- VEG : Vecteur indiquant les numéros des variables qui sont dans la matrice de travail
- DIMB : Dimension de la matrice de travail.
- FEOD : Vecteur contenant le débit à un nœud (quand celui-ci est imposé)
- FICHE-ALPHA : fichiers des perméabilités
- PERMEABILITE : Variable du type fiche-alpha.
- LOCE : Fichier des connectivités
- ELT : Variable du type locc
- Typelem : type d'élément (23, 42, 49, 36)
- classe : Indique la classe de perméabilité
- ALD : Alimentation distribuée
- Nœud : vecteur contenant les numéros des nœuds.
- CORG : fichier des coordonnées.
- COORD : Variable du type CORG

Une variable COORD comprend:

XXH : vecteur contenant les coordonnées
des éléments (x, y, z)

LRCH : (-1) ou (0)

VALH : Valeur de la charge si celle-ci est
connue

DEBSP : Valeur du débit imposé à un
nœud.

FICHEQ : fichier des équations éliminées

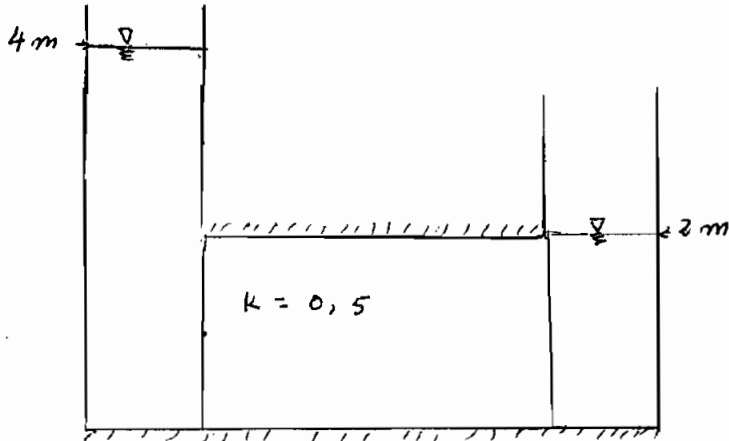
EQUOX : Variable du type FICHEQ

FICHSP : fichier des solutions

VALSP : Variable du type FICHSP.

TESTS DE VERIFICATION

Les deux premiers tests ont porté sur le problème physique suivant :



Solution analytique :

on appliquant la loi de darcy on obtient :

$$Q = kA \frac{\Delta h}{\Delta x}$$

avec $A = 2$
 $k = 0,5$
 $\Delta h = 2$
 $\Delta x = 4$

soit

$$Q = 0,5 \times 2 \times \frac{2}{4}$$

$$Q = 0,5 \text{ m}^3/\text{s}.$$

.....
 : DONNEES :
 :.....

1

Donnees generales

Nbr total de noeuds : 8
 Nbr total d'elements : 1
 Nbre de classes de permeabilite : 1

Permeabilite

CLASSE	K 1	K 2	K 3	K 4	K 5	K 6
1	0.5000	0.0000	0.0000	0.5000	0.0000	0.0000

Connectivite

Elt nx	Type	Classe	ALD	Nd1	Nd2	Nd3	Nd4	Nd5	Nd6	Nd7	Nd8	Nd9
1	48	1	0.0000	1	2	3	4	5	6	7	8	

COORdonnees et condition aux limites

Noeud nx	Xcoord	Ycoord	Zcoord	Hhcx	valh	Debit
1	0.00	0.00	0.00	-1	4.00	0.00
2	2.00	0.00	0.00	0	0.00	0.00
3	4.00	0.00	0.00	-1	2.00	0.00
4	4.00	1.00	0.00	-1	2.00	0.00
5	4.00	2.00	0.00	-1	2.00	0.00
6	2.00	2.00	0.00	0	0.00	0.00
7	0.00	2.00	0.00	-1	4.00	0.00
8	0.00	1.00	0.00	-1	4.00	0.00

.....
 : RESULTATS CALCULES :

Variable Nx	Debit calculee	Charge calculee
8	0.3333	4.0000
7	0.0833	4.0000
6	0.0000	3.0000
5	-0.0833	2.0000
4	-0.3333	2.0000
3	-0.0833	2.0000
2	0.0000	3.0000
1	0.0833	4.0000

Somme des debits sortants = -4.999999999999999E-001
 Somme des debits entrants = 5.000000000000002E-001
 Somme des debits = 2.33146835171283E-015

.....
 : DONNEES : 2
 :.....

Donnees generales

Nbr total de noeuds : 9
 Nbr total d'elements : 2
 Nbre de classes de permeabilite : 2

Permeabilite

CLASSE	K 1	K 2	K 3	K 4	K 5	K 6
1	0.5000	0.0000	0.0000	0.5000	0.0000	0.0000
2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Connectivite

Elt nx	Type	Classe	ALD	Nd1	Nd2	Nd3	Nd4	Nd5	Nd6	Nd7	Nd8	Nd9
1	49	1	0.0000	1	2	3	4	5	6	7	8	9
2	23	2	0.2500	7	8	1						

COORdonnees et condition aux limites

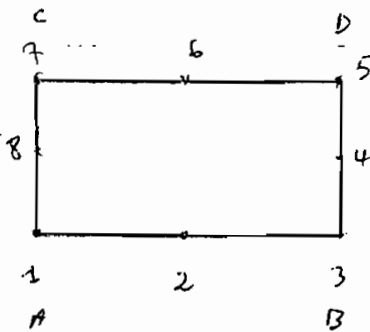
Noeud nx	Xcoord	Ycoord	Zcoord	Hhcx	valh	Debit
1	0.00	0.00	0.00	0	0.00	0.00
2	2.00	0.00	0.00	0	0.00	0.00
3	4.00	0.00	0.00	-1	2.00	0.00
4	4.00	1.00	0.00	-1	2.00	0.00
5	4.00	2.00	0.00	-1	2.00	0.00
6	2.00	2.00	0.00	0	0.00	0.00
7	0.00	2.00	0.00	0	0.00	0.00
8	0.00	1.00	0.00	0	0.00	0.00
9	2.00	1.00	0.00	0	0.00	0.00

: RESULTATS CALCULES :

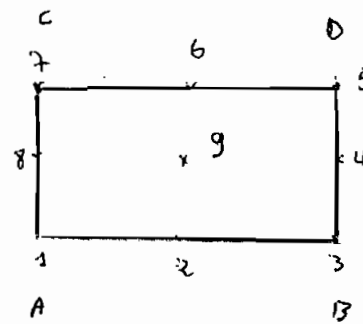
Variable Nx Debit calculee Charge calculee

Variable Nx	Debit calculee	Charge calculee
1	0.0833	4.0000
8	0.3333	4.0000
7	0.0833	4.0000
9	0.0000	3.0000
6	0.0000	3.0000
5	-0.0833	2.0000
4	-0.3333	2.0000
3	-0.0833	2.0000
2	0.0000	3.0000

Somme des debits sortants = -5.0000000000000002E-001
 Somme des debits entrants = 5.0000000000000000E-001
 Somme des debits = -1.66533453693773E-015



Type 48



Type 49

Charge imposee :

noeuds 7, 8, 5 : $h = 4\text{ m}$

noeud 3, 4, 5 : $h = 2\text{ m}$

$q_s = 0$ sur AB et CD

charge imposee

$h = 2\text{ m}$ aux noeuds 3, 4 et 5

$q_s = 0,25\text{ m}^3/\text{s}\cdot\text{m}$ sur AC

$k = 0,5\text{ m/s}$

.....
 : DONNEES : 3
 :

Donnees generales

Nbr total de noeuds : 21
 Nbr total d'elements : 4
 Nbre de classes de permeabilite : 1

Permeabilite

CLASSE	K 1	K 2	K 3	K 4	K 5	K 6
1	1.0000	0.0000	0.0000	1.0000	0.0000	0.0000

Connectivite

Elt nx	Type	Classe	ALD	Nd1	Nd2	Nd3	Nd4	Nd5	Nd6	Nd7	Nd8	Nd9
1	48	1	0.0000	1	2	3	7	11	10	9	8	
2	48	1	0.0000	3	4	5	6	13	12	11	7	
3	48	1	0.0000	9	10	11	15	19	18	17	16	
4	48	1	0.0000	11	12	13	14	21	20	19	15	

COORdonnees et condition aux limites

Noeud nx	Xcoord	Ycoord	Zcoord	Hhcx	valh	Debit
1	0.00	0.00	0.00	-1	0.00	0.0000
2	250.00	0.00	0.00	0	0.00	0.0000
3	500.00	0.00	0.00	0	0.00	0.0000
4	750.00	0.00	0.00	0	0.00	0.0000
5	1000.00	0.00	0.00	0	0.00	0.0000
6	1000.00	250.00	0.00	0	0.00	0.0000
7	500.00	250.00	0.00	0	0.00	0.0000
8	0.00	250.00	0.00	-1	0.25	0.0000
9	0.00	500.00	0.00	-1	0.50	0.0000
10	250.00	500.00	0.00	0	0.00	0.0000
11	500.00	500.00	0.00	0	0.00	0.0000
12	750.00	500.00	0.00	0	0.00	0.0000

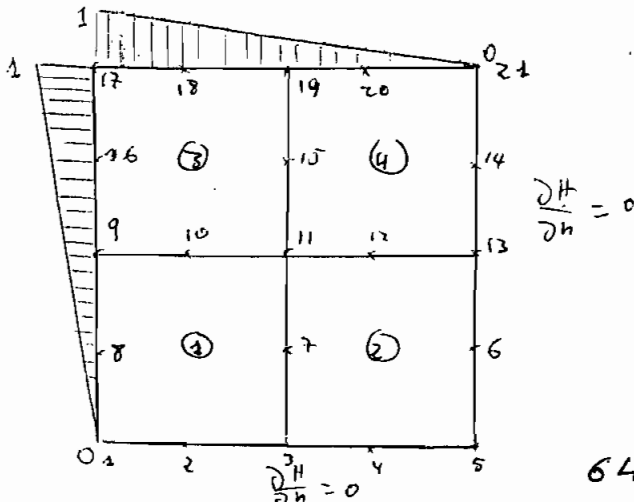
13	1000.00	500.00	0.00	0	0.00	0.0000
14	1000.00	750.00	0.00	0	0.00	0.0000
15	500.00	750.00	0.00	0	0.00	0.0000
16	0.00	750.00	0.00	-1	0.75	0.0000
17	0.00	1000.00	0.00	-1	1.00	0.0000
18	250.00	1000.00	0.00	-1	0.75	0.0000
19	500.00	1000.00	0.00	-1	0.50	0.0000
20	750.00	1000.00	0.00	-1	0.25	0.0000
21	1000.00	1000.00	0.00	-1	0.00	0.0000

: RESULTATS CALCULES :

Variable Nx Debit calculee Charge calculee

15	0.000000	0.4530
19	0.010575	0.5000
20	-0.095582	0.2500
21	-0.193153	0.0000
14	0.000000	0.2547
13	0.000000	0.3410
12	0.000000	0.3540
11	0.000000	0.4184
16	0.191668	0.7500
17	0.172983	1.0000
18	0.191668	0.7500
10	0.000000	0.4530
9	0.010575	0.5000
7	0.000000	0.3540
6	0.000000	0.3552
5	0.000000	0.3386
4	0.000000	0.3552
3	0.000000	0.3410
8	-0.095582	0.2500
2	0.000000	0.2547
1	-0.193153	0.0000

Somme des debits sortants = -5.7746923264E-01
Somme des debits entrants = 5.7746923261E-01
Somme des debits = -2.2737367544E-11



K = 1 m/s
cote 2000 m
h = 0 aux noeds 21 et 1
h = 1 aux noed 17

.....
 : DONNEES :
 :.....

(4)

Donnees generales

Nbr total de noeuds : 65
 Nbr total d'elements : 16
 Nbre de classes de permeabilite : 1

Permeabilite

CLASSE	K 1	K 2	K 3	K 4	K 5	K 6
1	0.0050	0.0000	0.0000	0.0050	0.0000	0.0000

Connectivite

Elt nx	Type	Classe	ALD	Nd1	Nd2	Nd3	Nd4	Nd5	Nd6	Nd7	Nd8	Nd9
1	48	1	0.0000	1	2	3	13	17	16	15	14	
2	48	1	0.0000	3	4	5	12	19	18	17	13	
3	48	1	0.0000	5	6	7	11	21	20	19	12	
4	48	1	0.0000	7	8	9	10	23	22	21	11	
5	48	1	0.0000	15	16	17	27	31	30	29	28	
6	48	1	0.0000	17	18	19	26	33	32	31	27	
7	48	1	0.0000	19	20	21	25	35	34	33	26	
8	48	1	0.0000	21	22	23	24	37	36	35	25	
9	48	1	0.0000	29	30	31	41	45	44	43	42	
10	48	1	0.0000	31	32	33	40	47	46	45	41	
11	48	1	0.0000	33	34	35	39	49	48	47	40	
12	48	1	0.0000	43	44	45	55	59	58	57	56	
13	48	1	0.0000	35	36	37	38	51	50	49	39	
14	48	1	0.0000	45	46	47	54	61	60	59	55	
15	48	1	0.0000	47	48	49	53	63	62	61	54	
16	48	1	0.0000	49	50	51	52	65	64	63	53	

COORdonnees et condition aux limites

Noeud nx	Xcoord	Ycoord	Zcoord	Hhex	valh	Debit
1	0.00	0.00	0.00	-1	600.00	0.00
2	125.00	0.00	0.00	-1	600.00	0.00
3	250.00	0.00	0.00	-1	600.00	0.00

4	375.00	0.00	0.00	-1	600.00	0.00
5	500.00	0.00	0.00	-1	600.00	0.00
6	625.00	0.00	0.00	-1	600.00	0.00
7	750.00	0.00	0.00	-1	600.00	0.00
8	875.00	0.00	0.00	-1	600.00	0.00
9	1000.00	0.00	0.00	-1	600.00	0.00
10	1000.00	125.00	0.00	-1	600.00	0.00
11	750.00	125.00	0.00	0	0.00	0.00
12	500.00	125.00	0.00	0	0.00	0.00
13	250.00	125.00	0.00	0	0.00	0.00
14	0.00	125.00	0.00	-1	600.00	0.00
15	0.00	250.00	0.00	-1	600.00	0.00
16	125.00	250.00	0.00	0	0.00	0.00
17	250.00	250.00	0.00	0	0.00	0.00
18	375.00	250.00	0.00	0	0.00	0.00
19	500.00	250.00	0.00	0	0.00	0.00
20	625.00	250.00	0.00	0	0.00	0.00
21	750.00	250.00	0.00	0	0.00	0.00
22	875.00	250.00	0.00	0	0.00	0.00
23	1000.00	250.00	0.00	-1	600.00	0.00
24	1000.00	375.00	0.00	-1	600.00	0.00
25	750.00	375.00	0.00	0	0.00	0.00
26	500.00	375.00	0.00	0	0.00	0.00
27	250.00	375.00	0.00	0	0.00	0.00
28	0.00	375.00	0.00	-1	600.00	0.00
29	0.00	500.00	0.00	-1	600.00	0.00
30	125.00	500.00	0.00	0	0.00	0.00
31	250.00	500.00	0.00	0	0.00	0.00
32	375.00	500.00	0.00	0	0.00	0.00
33	500.00	500.00	0.00	0	0.00	-0.0125
34	625.00	500.00	0.00	0	0.00	0.00
35	750.00	500.00	0.00	0	0.00	0.00
36	875.00	500.00	0.00	0	0.00	0.00
37	1000.00	500.00	0.00	-1	600.00	0.00
38	1000.00	625.00	0.00	-1	600.00	0.00
39	750.00	625.00	0.00	0	0.00	0.00
40	500.00	625.00	0.00	0	0.00	0.00
41	250.00	625.00	0.00	0	0.00	0.00
42	0.00	625.00	0.00	-1	600.00	0.00
43	0.00	750.00	0.00	-1	600.00	0.00
44	125.00	750.00	0.00	0	0.00	0.00
45	250.00	750.00	0.00	0	0.00	0.00
46	375.00	750.00	0.00	0	0.00	0.00
47	500.00	750.00	0.00	0	0.00	0.00
48	625.00	750.00	0.00	0	0.00	0.00
49	750.00	750.00	0.00	0	0.00	0.00
50	875.00	750.00	0.00	0	0.00	0.00
51	1000.00	750.00	0.00	-1	600.00	0.00
52	1000.00	875.00	0.00	-1	600.00	0.00
53	750.00	875.00	0.00	0	0.00	0.00
54	500.00	875.00	0.00	0	0.00	0.00
55	250.00	875.00	0.00	0	0.00	0.00
56	0.00	875.00	0.00	-1	600.00	0.00
57	0.00	1000.00	0.00	-1	600.00	0.00
58	125.00	1000.00	0.00	-1	600.00	0.00
59	250.00	1000.00	0.00	-1	600.00	0.00
60	375.00	1000.00	0.00	-1	600.00	0.00

61	500.00	1000.00	0.00	-1	600.00	0.00
62	625.00	1000.00	0.00	-1	600.00	0.00
63	750.00	1000.00	0.00	-1	600.00	0.00
64	875.00	1000.00	0.00	-1	600.00	0.00
65	1000.00	1000.00	0.00	-1	600.00	0.00

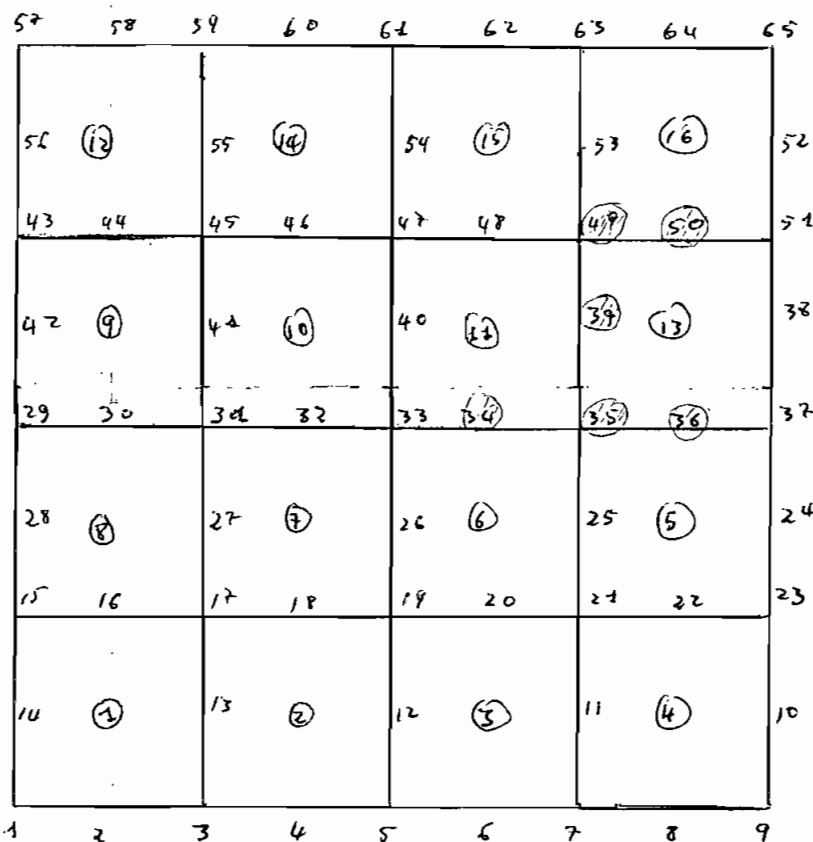
: RESULTATS CALCULES :

Variable Nx Debit calculee Charge calculee

53	0.0000	599.9244
63	0.0004	600.0000
64	0.0003	600.0000
65	-0.0000 ⁰⁹	600.0000
52	0.0003	600.0000
51	0.0004	600.0000
50	0.0000	599.9244
49	0.0000	599.8452
54	0.0000	599.8822
61	0.0007	600.0000
62	0.0006	600.0000
48	0.0000	599.7133
47	0.0000	599.7325
55	0.0000	599.9244
59	0.0004	600.0000
60	0.0006	600.0000
46	0.0000	599.7133
45	0.0000	599.8452
39	0.0000	599.7133
38	0.0006	600.0000
37	0.0007	600.0000
36	0.0000	599.8822
35	0.0000	599.7325
56	0.0003	600.0000
57	-0.0000 ⁰⁹	600.0000
58	0.0003	600.0000
44	0.0000	599.9244
43	0.0004	600.0000
40	0.0000	599.3511
34	0.0000	599.3511
33	-0.0125	598.5820
41	0.0000	599.7133
32	0.0000	599.3511
31	0.0000	599.7325
42	0.0006	600.0000
30	0.0000	599.8822
29	0.0007	600.0000
25	0.0000	599.7133
24	0.0006	600.0000
23	0.0004	600.0000
22	0.0000	599.9244
21	0.0000	599.8452

26	0.0000	599.3511
20	0.0000	599.7133
19	0.0000	599.7325
27	0.0000	599.7133
18	0.0000	599.7133
17	0.0000	599.8452
28	0.0006	600.0000
16	0.0000	599.9244
15	0.0004	600.0000
11	0.0000	599.9244
10	0.0003	600.0000
9	-0.000009	600.0000
8	0.0003	600.0000
7	0.0004	600.0000
12	0.0000	599.8822
6	0.0006	600.0000
5	0.0007	600.0000
13	0.0000	599.9244
4	0.0006	600.0000
3	0.0004	600.0000
14	0.0003	600.0000
2	0.0003	600.0000
1	-0.000009	600.0000

Somme des debits sortants = -1.25369921688279E-002
 Somme des debits entrants = 1.25369921688188E-002
 Somme des debits = -9.14893161230168E-015



Largeur 1000 m
 $k = 0,005 \text{ m/s}$
 charge à tous les nœuds
 du contour: $h = 600 \text{ m}$.
 Débit imposé au nœud
 33 : $Q = -0,0125 \text{ m}^3/\text{s}$

Nous utilisons pour la solution analytique la théorie de Dupuit de l'écoulement permanent vers un puits en nappe captive.

$$\text{Soit } s_0 = \frac{Q}{2\pi T h_0} \ln \frac{R}{r_0}$$

Porchet propose de prendre

$$\frac{R}{r_0} = 100 \quad \text{soit} \quad \frac{\ln \frac{R}{r_0}}{\pi} = 1,5$$

$$\text{avec } Q = 0,0125 \text{ m}^3/\text{s}$$

$$T = 0,005 \text{ m}^2/\text{s}$$

$$\Rightarrow s_0 = \frac{0,0125}{2 \times 0,005} \times 1,5 = 1,875$$

ce qui donne $h = H - s_0$

$$h = 600 - 1,875 = 598,125 \text{ m.}$$

ce qui donne un pourcentage d'erreur :

$$\Delta = \frac{|598,125 - 598,582|}{598,582} = 0,07\%$$

PROCEDURE DEBUT;

Var foostext;

Begin clrscr;Assign(foo,'con');Rewrite(foo);

Writeln('%%')

Writeln('

Writeln('

REPUBLIQUE DU SENEGAL
ECOLE POLYTECHNIQUE DE THIES
DEPARTEMENT du Genie Civil

Writeln('

Writeln('

Writeln('

Writeln('

Writeln('

PROJET DE FIN D'ETUDES

Writeln('

Writeln('

Titre : SIMULATION DE L'ECOULEMENT EN MILIEU POREUX SATURE
PAR LA METHODE DES ELEMENTS FINIS.

Writeln('

Writeln('

Writeln('

Writeln('

Writeln('

Auteur : Ametepe Yayrato NENDONENE

Writeln('

Directeur : Gerard SOUMA

Writeln('

Professeur d"hydrogeologie

Writeln('

Co-directeur: Amadou SARR

Writeln('

Professeur d"hydraulique

Writeln('

Writeln('%%')

Writeln;Write(foo,chr(27),'[5m');Write(foo,chr(27),'[7m');

Write(foo,'

Bienvenue sur NENDONENE Taper <Return> pour continuer ');

Write(foo,chr(27),'[0m');

Read;

```

program NENO_EPT(input,output);
  type
    inter1=array[1..30] of integer;
    typecoord=array[1..227] of real;
    typemat=array[1..27,1..27] of real;
    inter=array [1..27] of real;
    Element=RECORD
      Type1mt,
      Classe:integer;
      Ald:real;
      noeud :array[1..27] of integer;
    end;
    Rat=Record
      xxn:array[1..3] of real;
      Hhcx:integer;
      Valh:real;
      valqsp:real;
    end;
    Equation=Record
      coefx:inter;
      coeff:real;
      place:integer;
      numero:integer;
      long:integer;
      cxx:integer;
      lhh:real;
      debp:real;
    end;
    Solution = Record
      vari:integer;
      solq:real;
      solh:real;
    end;
    Alpha=RECORD
      Classperm:integer;
      K:Array[1..6] of real;
    End;

```

```

var
  idi,idr:array[1..100] of integer;
  elt:element;
  coord:rat;
  nbpt,nbtelt,n,dimb,pel,e,nbtnd,i,j,k,kp,ip,np:integer;
  fek,fn,fnds,fndt,fndu:inter;
  nbreperm,dimcek,nbnd,dimelt,nbelt,ind,ntn,nte:integer;
  somdbneg,somdbpos,total,qs,deter:real;
  gek,perm,bb,so,a,t,c,b:typemat;
  xn,yn,zn:typecoord;
  Ficheq:File of Equation;
  LOCE:File of element;
  CORG:file of rat;
  Fiche_alpha:file of alpha;
  Equax:Equation;
  fichsol:File of solution;
  Valsol:solution;
  permeabilite:alpha;
  posi,hhcxx,cek,vek,Veg:inter1;
  valqspp,fegd,valhh,feg:inter;
  geg:array[1..30,1..30] of real;
  Option1,option2,option3,option4:integer;
  label 50,100,200,300,400;

```

```
{$I NENOSUIT.PAS}
```

```
{$I DEBUT.PAS}
```



```

procedure FN23(var fn,fnds:inter; s:real);
  var
    a:real;
  begin
    a:=s*s;

    fn[1]:=0.5*(a-s);
    fn[2]:=1.0-a;
    fn[3]:=0.5*(a+s);

    fnds[1]:=s-0.5;
    fnds[2]:=-s-s;
    fnds[3]:=s+0.5;
  end;

```

```

procedure FN36(var fn,fnds,fndt:inter; al,b1,cl:real);
  var
    a,b,c: real;
    as,at,bs,bt,cs,ct: real;
  begin
    a:=al*b1*2;
    b:=b1*cl*2;
    c:=cl*al*2;
    as:=b1+b1;
    at:=al+al;
    bs:=-as;
    bt:=cl+cl-at;
    ct:=-at;
  end;

```

```

fn[1]:=a1-c-a;
fn[2]:=a+a;
fn[3]:=b1-a-b;
fn[4]:=b+b;
fn[5]:=c1-b-c;
fn[6]:=c+c;

fnds[1]:=1-cs-as;
fnds[2]:=as+as;
fnds[3]:=0;
fnds[4]:=bs+bs;
fnds[5]:=-1-bs-cs;
fnds[6]:=cs+cs;

fndt[1]:=0;
fndt[2]:=at+at;
fndt[3]:=1-at-bt;
fndt[4]:=bt+bt;
fndt[5]:=-1-bt-ct;
fndt[6]:=ct+ct;
end;
procedure FN48(var fn,fnds,fndt:inter; s,t:real);
var
  a,b,c,d:real;
  cs,ct,sm,sp,tm,tp,sq,tq:real;
  as,at,bs,bt,cc,tt,ds,dt:real;
  fndu:inter;
begin
  cs:=0.5*s;
  ct:=0.5*t;
  sm:=0.5-cs;
  sp:=0.5+cs;
  tm:=0.5-ct;
  tp:=0.5+ct;
  sq:=0.5-s*cs;
  tq:=0.5-t*ct;

```

```

a:=sq*tm;
b:=sp*tq;
c:=sq*tp;
d:=sm*tq;
as:=-s*tm;
at:=-0.5*sq;
bs:=0.5*tq;
bt:=-t*sp;
cc:=-s*tp;
tt:=0.5*sq;
ds:=-0.5*tq;
dt:=-t*sm;

for i:=1 to 8 do
begin
fn[i]:=0;
fnds[i]:=0;
fndt[i]:=0;
fndu[i]:=0;
end;
fn[1]:=sm*tm-d-a;
fn[2]:=a+a;
fn[3]:=sp*tm-a-b;
fn[4]:=b+b;
fn[5]:=sp*tp-b-c;
fn[6]:=c+c;
fn[7]:=sm*tp-c-d;
fn[8]:=d+d;
fnds[1]:=-0.5*tm-ds-as;
fnds[2]:=as+as;
fnds[3]:=0.5*tm-as-bs;
fnds[4]:=bs+bs;
fnds[5]:=0.5*tp-bs-cc;
fnds[6]:=cc+cc;
fnds[7]:=-0.5*tp-cc-ds;
fnds[8]:=ds+ds;
fndt[1]:=-0.5*sm-dt-at;

```

```

    fndt[2]:=at+at;
    fndt[3]:=-0.5*sp-at-bt;
    fndt[4]:=bt+bt;
    fndt[5]:=0.5*sp-bt-tt;
    fndt[6]:=tt+tt;
    fndt[7]:=0.5*sm-tt-dt;
    fndt[8]:=dt+dt;
    end;

```

```

procedure FN49(var fn,fnds,fndt:inter; s,t:real);

```

```

    var

```

```

        a,b,sqm,sqp,sq,tqm,tqp,tq:real;
        dsqm,dsqp,dsq,dtqm,dtqp,dtq:real;

```

```

begin

```

```

    a:=s*s;
    b:=t*t;
    sqm:=0.5*(a-s);
    sqp:=0.5*(a+s);
    sq:=1-a;
    tqm:=0.5*(b-t);
    tqp:=0.5*(b+t);
    tq:=1-b;
    dsqm:=s-0.5;
    dsqp:=s+0.5;
    dsq:=-s-s;
    dtqm:=t-0.5;
    dtqp:=t+0.5;
    dtq:=-t-t;

```

```

    fn[1]:=sqm*tqm;
    fn[2]:=sq*tqm;
    fn[3]:=sqp*tqm;
    fn[4]:=sqp*tq;
    fn[5]:=sqp*tqp;
    fn[6]:=sq*tqp;
    fn[7]:=sqm*tqp;
    fn[8]:=sqm*tq;

```

```

fn[9]:=sq*tq;

fnds[1]:=dsqm*tqm;
fnds[2]:=dsq*tqm;
fnds[3]:=dsqp*tqm;
fnds[4]:=dsqp*tq;
fnds[5]:=dsqp*tqp;
fnds[6]:=dsq*tqp;
fnds[7]:=dsqm*tqp;
fnds[8]:=dsqm*tq;
fnds[9]:=dsq*tq;

fndt[1]:=sqm*dtqm;
fndt[2]:=sq*dtqm;
fndt[3]:=sqp*dtqm;
fndt[4]:=sqp*dtq;
fndt[5]:=sqp*dtqp;
fndt[6]:=sq*dtqp;
fndt[7]:=sqm*dtqp;
fndt[8]:=sqm*dtq;
fndt[9]:=sq*dtq;
end;

```

```

procedure FN410(var fn,fnds,fndt,fndu:inter; a1,b1,c1,d1:real);
var
  a2,b2,c2,d2:real;
begin
  a2:=a1+a1;
  b2:=b1+b1;
  c2:=c1+c1;
  d2:=d1+d1;

  fn[1]:=a1*(a2-1);
  fn[2]:=a2*b2;
  fn[3]:=b1*(b2-1);
  fn[4]:=b2*c2;

```

```
fn[5]:=c1*(c2-1);
fn[6]:=c2*a2;
fn[7]:=a2*d2;
fn[8]:=b2*d2;
fn[9]:=c2*d2;
fn[10]:=d1*(d2-1);
```

```
fnds[1]:=a2+a2-1;
fnds[2]:=b2+b2;
fnds[3]:=0;
fnds[4]:=0;
fnds[5]:=0;
fnds[6]:=c2+c2;
fnds[7]:=d2+d2-a2-a2;
fnds[8]:=-b2-b2;
fnds[9]:=-c2-c2;
fnds[10]:=-d2-d2+1;
```

```
fndt[1]:=0;
fndt[2]:=a2+a2;
fndt[3]:=b2+b2-1;
fndt[4]:=c2+c2;
fndt[5]:=0;
fndt[6]:=0;
fndt[7]:=-a2-a2;
fndt[8]:=d2+d2-b2-b2;
fndt[9]:=-c2-c2;
fndt[10]:=-d2-d2+1;
```

```
fndu[1]:=0;
fndu[2]:=0;
fndu[3]:=0;
fndu[4]:=b2+b2;
fndu[5]:=c2+c2-1;
fndu[6]:=a2+a2;
fndu[7]:=-a2-a2;
fndu[8]:=-b2-b2;
```

```
fndu[9]:=d2+d2-c2-c2;  
fndu[10]:=-d2-d2+1;  
end;
```

```
{ # SOUS PROGRAMME FAISANT LE CHOIX # }  
{ # DU NOMBRE DE NOEUD ET DE LA DIMENSION # }  
{ # DE L'ELEMENT SUIVANT LE TYPE DE L'ELEMENT # }
```

```
Procedure CHOIX2 ;
```

```
  BEGIN
```

```
    Case Elt.typelmt of
```

```
      23 : begin  
          dimelt:=1;  
          nbnd:=3;  
        end;
```

```
      36 : begin  
          dimelt:=2 ;  
          nbnd:=6;  
        end;
```

```
      48 : begin  
          dimelt:=2 ;  
          nbnd:=8;  
        end;
```

```
      49 : begin  
          dimelt:=2 ;  
          nbnd:=9;  
        end;
```

```
     410: begin  
          dimelt:=3 ;  
          nbnd:=10;  
        end;
```

```
  end;
```

```
END;
```

```

    { # CALCUL DU VECTEUR 'VEK' CORRESPONDANT AU # }
    { #          NUMERO DU NOEUD          # }

```

```

Procedure VECVEK;

```

```

  BEGIN

```

```

    CHOIX2;

```

```

    WITH Elt do

```

```

      For i:=1 to nbnd do

```

```

        vek[i]:=noeud[i];

```

```

      END;

```

```

    { * CALCUL DES CODES ATTRIBUES AUX NOEUDS * }

```

```

Procedure VECTIDI;

```

```

  BEGIN

```

```

    For i:=1 to nbtnd do

```

```

      begin

```

```

        idi[i]:=0;

```

```

        idr[i]:=0;

```

```

      end;

```

```

    ASSIGN(loce, 'b:NENO.DAT');

```

```

    RESET(loce);

```

```

    WITH elt do

```

```

      begin

```

```

        for i:=1 to nbtelt do

```

```

          begin

```

```

            k:=i-1;

```

```

            SEEK(loce,k);

```

```

            READ(loce,elt);

```

```

            CHOIX2;

```

```

            for j:=1 to nbnd do

```

```

              idi[noeud[j]]:=idi[noeud[j]]+1;

```

```

            end;

```

```

          end;

```

```

    CLOSE(loce);

```

```

    For i:=1 to nbtnd do

```

```

      idr[i]:=idi[i];

```


END;

{ * CALCUL DES CODES ATTRIBUES AUX NOEUDS * }

Procedure VECTCEK;

BEGIN

CHOIX2;

For n:=1 to nbnd do

begin

if idi[vek[n]]=1

then

cek[n]:=1

else

begin

if idi[vek[n]]=idr[vek[n]]

then

begin

cek[n]:=idi[vek[n]];

idr[vek[n]]:=idr[vek[n]]-2;

end

else

begin

if idr[vek[n]]=0

then

cek[n]:=0

else

begin

cek[n]:=-1;

idr[vek[n]]:=idr[vek[n]]-1;

end;

end;

end;

end;

END;

```

{ # LECTURE SUR FICHER 'ficheq' APRES ELIMINATION # }
{ # D'UNE VARIABLE QUI APPARAIT POUR LA DERNIERE FOIS # }
{ #      DANS LA MATRICE DE TRAVAIL      # }

```

```

procedure ENREG(var equax:equation);
  var
    equaxi:equation;
    l:integer;

  BEGIN
    with equax do
      begin
        if e=1
          then
            begin
              ASSIGN(ficheq,'NENO.EQT');
              REWRITE(ficheq);
              write(ficheq,equax);
              CLOSE(ficheq);
            end

          else
            begin
              ASSIGN(ficheq,'NENO.EQT');
              RESET(ficheq);
              while not eof(ficheq) do
                read(ficheq,equaxi);
                write(ficheq,equax);
              CLOSE(ficheq);
            end;
          for l:=1 to dimb do
            coefx[l]:=0;
            coeff:=0;
            numero:=0;

```

```

        place:=0;
        long:=0;
    end;
END;

```

```

{ # RECHERCHE DES EQUATIONS DANS LE FICHIER # }
{ # 'ficheq' POUR LE CALCUL DES INCONNUES DU # }
{ # PROBLEME # }

```

```

Procedure RECHERCHE(Var equax:equation;var np:integer);

```

```

    var kk:integer;

```

```

    BEGIN

```

```

        kk:=np-1;

```

```

        ASSIGN(ficheq,'NENO.EQT');

```

```

        RESET(ficheq);

```

```

        SEEK(ficheq,kk);

```

```

        Read(ficheq,equax);

```

```

        CLOSE(ficheq);

```

```

    END;

```

```

{ # LECTURE SUR FICHIER 'fichsol' DES RESULTATS # }
{ # CALCUL # }

```

```

Procedure RESULTAT(Var valsol:solution);

```

```

    VAR

```

```

        valsoli:solution;

```

```

    BEGIN

```

```

        with valsol do

```

```

            begin

```

```

                if Np=nbtd

```

```

                THEN

```

```

                    begin

```

```

                        ASSIGN(fichsol,'NENO.SVL');

```

```

        REWRITE(fichsol);
        with valsol do
            write(fichsol,valsol);
        CLOSE(fichsol);
    end
ELSE
    begin
        ASSIGN(fichsol,'NENO.SVL');
        RESET(fichsol);
        while not eof(fichsol) do
            read(fichsol,valsoli);
            write(fichsol,valsol);
        CLOSE(fichsol);
    end;
    vari:=0;
    solh:=0;
    solq:=0;
end;

```

END;

```

{ # IMPRESSION DES RESULTATS SUR IMPRIMANTE OU # }
{ #          SUR ECRAN SUIVANT L'OPTION          # }

```

Procedure IMPRESSION1 ;

```

BEGIN
    ASSIGN(fichsol,'NENO.SVL');
    ASSIGN(loce,'B:neno.dat');
    ASSIGN(corg,'B:neno.cor');
    ASSIGN(fiche_alpha,'B:nenoperm.dat');
    RESET(fiche_alpha);
    writeln(lst,' .....');
    writeln(lst,' : DONNEES :');
    writeln(lst,' :.....:');
    writeln(lst,' ');writeln(lst,' ');
    writeln(lst,' Donnees generales ');

```

```

writeln(lst,'      -----');
writeln(lst,' ');
writeln(lst,' Nbr total de noeuds           : ',nbtnd:3)
writeln(lst,' Nbr total d"elements         : ',nbtelt:3)
writeln(lst,' Nbre de classes de permeabilite : ',nbreperm)
writeln(lst);
writeln(lst,'      Permeabilite ');
writeln(lst,'      ----- ');
writeln(lst,' ');
With elt do
With coord do
With permeabilite do
begin
Write(lst,'      CLASSE ');
For i:=1 to 6 do
    Write(lst,'    K ',i,' ');
Writeln(lst);
For i:=1 to 80 do
write(lst,'_');
writeln(lst,' ');
writeln(lst,' ');
for I:=1 to nbreperm do
    begin
        SEEK(fiche_alpha,i-1);
        READ(fiche_alpha,permeabilite);
        write(lst,'      ',classperm,' ');
        for J:=1 to 6 do
            begin
                write(lst,'    ',k[J]:7:4,' ');
            end;
        writeln(lst,' ');
    end;
Close(fiche_alpha);
writeln(lst,' ');writeln(lst,' ');writeln(lst,' ');
RESET(loce);
writeln(lst,'      Connectivite ');
writeln(lst,'      ----- ');

```

```

writeln(lst,' ');
write(lst,' Elt nx ');
write(lst,' Type ');
write(lst,' Classe ');
write(lst,' ALD      ');
for i:=1 to 9 do
    write(lst,'Nd',i,' ');
writeln(lst,' ');
for i:=1 to 80 do
write(lst,'-');
writeln(lst,' ');
for i:=1 to nbtelt do
begin
    Seek(Loce,i-1);
    Read(Loce,Elt);
    Write(lst,' ',i:3,' ');
    Write(lst,' ',Typelmt:2);
    Write(lst,' ',Classe:2);
    write(lst,' ',ALD:8:4,' ');
    CHOIX2;
    for j:=1 to nbnd do
        write(lst,' ',noeud[j]:3);
    writeln(lst,' ');
end;
RESET(corg);
writeln(lst,' ');writeln(lst,' ');writeln(lst,' ');
writeln(lst,' COORdonnees et condition aux limites ');
writeln(lst,' ----- ');
writeln(lst,' ');
write(lst,' Noeud nx ');
write(lst,' Xcoord ');
write(lst,' Ycoord ');
write(lst,' Zcoord ');
write(lst,' Hhcx ');
write(lst,' valh ');
write(lst,' Debit ');
writeln(lst,' ');

```

```

for i:=1 to 80 DO
    write(lst,'-');
writeln(lst,' ');
For i:=1 to nbtnd do
    begin
        SEEK(Corg,i-1);
        READ(Corg,coord);
        write(lst,'      ',i:3,' ');
        for J:=1 to 3 do
            write(lst,' ',xxn[J]:8:2,'');
        write(lst,' ',hhcx:2,' ');
        write(lst,' ',valh:8:2,' ');
        write(lst,' ',valqsp:8:2,' ');
        writeln(lst,' ');
    end;
    writeln(lst,' ');
CLOSE(corg) ;
RESET(fichsol);
writeln(lst,' ----- ');
writeln(lst,' :   RESULTATS CALCULES   ');
writeln(lst,' ----- ');
writeln(lst);
write(lst,'Variable Nx ');
write(lst,' Debit calculee ');
write(lst,' Charge calculee ');
writeln(lst,' ');
for i:=1 to 80 do
    write(lst,'-');
writeln(lst,' ');
writeln(lst,' ');
for ip:=1 to nbtnd do
    begin
        SEEK(fichsol,ip-1);
        READ(fichsol,valsol);
        With valsol do
            begin
                write(lst,' ',vari:3,' ');

```

```

        write(lst, ' ', solq:10:4, ' ');
        write(lst, ' ', solh:10:4, ' ');
    end;
    writeln(lst, ' ');
end;
CLOSE(fichsol);
writeln(lst, ' ');
writeln(lst, ' Somme des debits sortants = ', somdbneg);
writeln(lst, ' Somme des debits entrants = ', somdbpos);
writeln(lst, ' Somme des debits          = ', total);
end;

END;

```

Procedure IMPRESSION2;

```

BEGIN
    ASSIGN(fichsol, 'NENO.SVL');
    ASSIGN(loce, 'B:neno.dat');
    ASSIGN(corg, 'B:neno.cor');
    ASSIGN(fiche_alpha, 'B:nenoperm.dat');
    RESET(fiche_alpha);
    writeln('.....');
    writeln(' :   DONNEES   : ');
    writeln(' :.....: ');
    writeln;writeln;
    writeln(' Donnees generales ');
    writeln(' ----- ');
    writeln;
    writeln(' Nbr total de noeuds          : ', nbtdnd:3);
    writeln(' Nbr total d"elements         : ', nbtdelt:3);
    writeln(' Nbre de classes de permeabilite : ', nbreperm:3);
    writeln('   Permeabilite   ');
    writeln(' ----- ');
    writeln;

```



```

writeln;
With elt do
With coord do
With permeabilite do
begin
Write(' ');posi[1]:=whereX+1;
Write(' CLASSE ');posi[2]:=whereX+1;
For i:=1 to 6 do
    Write(' K',i,' ');posi[i+1]:=WhereX+1;
Writeln;
For i:=1 to 80 do
write('_');
writeln;
gotoXY(posi[1],whereY);
for I:=1 to nbreperm do
    begin
        SEEK(fiche_alpha,i-1);
        READ(fiche_alpha,permeabilite);
        write(' ',classperm:2);
        for J:=1 to 6 do
            begin
                gotoxy(posi[J+1],whereY);
                write(k[J]:7:4);
            end;
        writeln;
    end;
Close(fiche_alpha);
read;
writeln;writeln;writeln;
RESET(loce);
writeln(' Connectivit ');
writeln(' ----- ');
writeln;
write(' Elt nx ');posi[1]:=whereX+1;
write(' Type ');posi[2]:=whereX+1;
write(' Classe ');posi[3]:=whereX+1;
write(' ALD ');posi[4]:=whereX+1;

```

```

for i:=1 to 9 do
  begin
    write( ' Nd',i,' ');
    posi[i+4]:=whereX+1;
  end;
writeln;
for i:=1 to 80 do
write('-');
for i:=1 to nbtelt do
begin
  Seek(Loce,i-1);
  Read(Loce,Elt);
  Write(' ',i,' ');GotoXY(posi[1],WhereY);
  Write(Typelmt:3);GotoXY(posi[2],WhereY);
  Write(Classe:3);GotoXY(posi[3],WhereY);
  write(ALD:7:2);GotoXY(posi[4],whereY);
  CHOIX2;
  for j:=1 to nbnd do
    begin
      write(noeud[j]:3);
      gotoXY(posi[J+4],whereY);
    end;
  writeln;writeln;
end;
RESET(corg);
read;
writeln;writeln;writeln;
writeln(' COORDonnees et condition aux limites ');
writeln(' ----- ');
writeln;
write(' Noeud nx ');posi[1]:=whereX+1;
write(' Xcoord ');posi[2]:=whereX+1;
write(' Ycoord ');posi[3]:=whereX+1;
Write(' Zcoord ');posi[4]:=whereX+1;
write(' Hhcx ');posi[5]:=whereX+1;
write(' valh ');posi[6]:=whereX+1;
write(' Debit ');posi[7]:=whereX+1;

```

```

writeln;
for i:=1 to 80 DO
  write('-');
writeln;
For i:=1 to nbtnd do
  begin
    SEEK(Corg,i-1);
    READ(Corg,coord);
    write(' ',i);gotoxy(posi[1],whereY);
    for J:=1 to 3 do
      begin
        write(xxn[J]:7:4);
        gotoxy(posi[J+1],whereY);
      end;
    write(' ',hhcx:3) ;gotoxy(posi[5],whereY);
    write(valh:7:4);gotoxy(posi[6],whereY);
    write(valqsp:7:4);
    writeln;writeln;
  end;
  read;
  WRITELN;
CLOSE(corg) ;
RESET(fichsol);
writeln(' ----- ');
writeln(':   RESULTATS CALCULES   ');
writeln(' ----- ');
write('Variable Nx ');posi[1]:=whereX+1;
write('Debit calculee ');posi[2]:=whereX+1;
write('Charge calculee ');posi[3]:=whereX+1;
writeln;
for i:=1 to 80 do
  write('-');
for ip:=1 to nbtnd do
  begin
    SEEK(fichsol,ip-1);
    READ(fichsol,valsol);
    With valsol do

```

```

begin
    write(' ',vari:3);
    gotoxy(posi[1],whereY);
    write(solq:8:4);
    gotoxy(posi[2],wherey);
    write(solh:8:4);
    gotoxy(posi[3],wherey);
end;
writeln;
end;
CLOSE(fichsol);
end;
writeln;
writeln(' Somme des debits negatifs = ',sombdneg);
writeln(' Somme des debits positifs = ',sombdpos);
writeln(' somme des debits = ',total );
END;

```

```

{ # INVERSION ET CALCUL DU DETERMINANT DES MATRICES # }
{ # UTILISE POUR LE CALCUL DU TENSEUR METRIQUE # }
{ # CONTRAVARIANT ET SON DETERMINANT # }

```

```

procedure INVERMATRI(var a,b:typemat;var ind,n:integer;var deter:real);

```

```

label 1,2,3,4,6,7;

```

```

var

```

```

    i,j,l,k,m: integer;

```

```

    r,tb,signe: real;

```

```

    a1:array[1..6,1..6] of real;

```

```

begin

```

```

    ind:=0;

```

```

    deter:=1;

```

```

signe:=1;
for i:=1 to n do
  for j:=1 to n do
    begin
      a1[i,j]:=a[i,j];
      a1[i,n+j]:=0;
      b[i,j]:=0;
    end;
  for i:=1 to n do
    a1[i,n+i]:=1 ;

for k:=1 to n-1 do
  6:begin
    begin
      if abs(a1[k,k])=0
      then goto 1
      else
        deter:=deter*a1[k,k];
        for i:=k+1 to n do
          begin
            r:=a1[i,k]/a1[k,k];
            for j:=k+1 to 2*n do
              a1[i,j]:=a1[i,j]-r*a1[k,j];
            end
          end;
        goto 2;

  1:begin
    m:=k+1;
    4 : if abs(a1[m,k])=0
      then
        begin
          m:=m+1;
          if m=n+1
            then goto 3
            else goto 4 ;
        end
      end
    end

```

```

        else
            l:=2*n;
            for j:=k to l do
                begin
                    r:=a1[k,j];
                    a1[k,j]:=a1[m,j];
                    a1[m,j]:=r;
                end;
            goto 6;
        end;
2 : end;

begin
    if abs(a1[n,n])=0
        then goto 3
        else
            deter:=deter*a1[n,n];
            for j:=n downto 1 do
                for i:=n downto 1 do
                    begin
                        tb:=0;
                        for k:=n downto i+1 do
                            tb:=tb-b[k,j]*a1[i,k];
                        b[i,j]:=(tb+a1[i,n+j])/a1[i,i];
                    end;
                end;
            end;
            goto 7;
        3: ind:=ind+1; exit;
7: end;

```

```

{ # CALCUL DES ELEMENTS DE LA BASE COVARIANTE # }

```

```

procedure jacob(var x: typecoord; var fnds, fndt, fndu: inter; var jp: inter; va

```

```

    var
        i: integer;

```

```

begin
  for i:=1 to 3 do
    jp[i]:=0;
  for i:=1 to nbnd do
    begin
      jp[1]:=x[i]*fnds[i]+jp[1];
      jp[2]:=x[i]*fndt[i]+jp[2];
      jp[3]:=x[i]*fndu[i]+jp[3];
    end;
  end;
end;

```

```
{ # CALCUL DU TENSEUR METRIQUE COVARIANT # }
```

```
procedure MATRIG(var js,g:typemat; var dimg:integer);
```

```
var
```

```
    i,j:integer;
```

```
    som,som1,som2,som3:real;
```

```
begin
```

```
  for i:=1 to dimg do
```

```
    begin
```

```
      som:=0;
```

```
      for j:=1 to 3 do
```

```
        som:=som+js[j,i]*js[j,i];
```

```
        g[i,i]:=som;
```

```
      end;
```

```
    som1:=0;
```

```
    som2:=0;
```

```
    som3:=0;
```

```
  for i:=1 to 3 do
```

```
    begin
```

```
      som1:=som1+js[i,1]*js[i,2];
```

```
      som2:=som2+js[i,1]*js[i,3];
```

```
      som3:=som3+js[i,2]*js[i,3];
```

```
    end;
```

```
  g[1,2]:=som1;
```

```

    g[2,1]:=som1;
    g[1,3]:=som2;
    g[3,1]:=som2;
    g[3,2]:=som3;
    g[2,3]:=som3;
end;

```

```
{ # CALCUL LE PRODUIT DE DEUX MATRICE # }
```

```
procedure PRODAB(var a,b,c:typemat;var nbla,nbcb,nblb:integer);
```

```
var
```

```

    k,i,j:integer;
    prod:real;
    ind2:integer;

```

```
begin
```

```
    ind2:=0;
```

```
    for i:=1 to nbla do
```

```
        begin
```

```
            for j:=1 to nbcb do
```

```
                begin
```

```
                    prod:=0;
```

```
                    for k:=1 to nblb do
```

```
                        prod:=prod+a[i,k]*b[k,j];
```

```
                        c[i,j]:=prod;
```

```
                    end;
```

```
            end;
```

```
    end;
```

```
{ # ~~~~~~# }
```

```
{ # CALCUL LA MATRICE DES GRADIENTS 'B' # }
```

```
{ # ~~~~~~# }
```

```
{ ~~~~~~ }
```

```
procedure MATRIB(var c,b:typemat; var fnds,fn dt,fndu:inter; var dimc,
```

```
var
```



```

        k,j:integer;

begin
    for k:=1 to dimc do
        for j:=1 to dimb do
            b[k,j]:=c[k,1]*fnds[j]+c[k,2]*fndt[j]+c[k,3]*fndu[j];
        end;
    end;

{ # TRANSPOSITION DES MATRICES UTILISE POUR # }
{ # LA TRASPOSITION DES MATRICE DES GRADIENTS # }

procedure TRANSPO(var an,at:typemat; var diml,dimc:integer);

var
    i,j:integer;

begin
    for i:=1 to diml do
        for j:=1 to dimc do
            at[j,i]:=an[i,j];
        end;
    end;

{ # CALCUL LA SOMME DE DEUX MATRICES UTILISE POUR # }
{ # LA SOMMATION DES MATRICES ELEMENTAIRES SUR LES # }
{ # POINTS DE GAUSS D'UN ELEMENT # }

procedure somatri(var ann,bnn,css:typemat; var diml,dimc:integer);

var
    i,j:integer;

begin
    for i:=1 to diml do
        for j:=1 to dimc do
            css[i,j]:=ann[i,j]+bnn[i,j];
        end;
    end;

```

```
{ # CALCUL TOUS LES ELEMENTS DE LA BASE COVARIANTE # }
```

```
procedure JACOBI(var js:typemat; var xn,yn,zn:typecoord; var fnds,fndt,fndu
```

```
var
```

```
  i:integer;
```

```
  jpp: inter;
```

```
begin
```

```
  jacob(xn,fnds,fndt,fndu,jpp,nbnd);
```

```
  for i:=1 to dimelt do
```

```
    begin
```

```
      js[1,i]:=jpp[i];
```

```
      jpp[i]:=0;
```

```
    end;
```

```
  jacob(yn,fnds,fndt,fndu,jpp,nbnd);
```

```
  for i:=1 to dimelt do
```

```
    begin
```

```
      js[2,i]:=jpp[i];
```

```
      jpp[i]:=0;
```

```
    end;
```

```
  jacob(zn,fnds,fndt,fndu,jpp,nbnd);
```

```
  for i:=1 to dimelt do
```

```
    begin
```

```
      js[3,i]:=jpp[i];
```

```
      jpp[i]:=0;
```

```
    end;
```

```
end;
```

```
{ # LES POINTS D'INTEGRATION POUR L'ELEMENT # }
```

```
{ #           1-D TYPE 23           # }
```

```
Procedure POINT23(var is:integer;var ws,ss:inter);
```

```
var
```

```
  s1,w1:inter;
```

```

Begin
  ws[is]:=0;
  ss[is]:=0;
  s1[1]:=-0.7745966692414834;
  s1[2]:=0;
  s1[3]:=-s1[1];
  w1[1]:=0.5555555555555555;
  w1[2]:=0.8888888888888888;
  w1[3]:=0.5555555555555555;
  ss[is]:=s1[is];
  ws[is]:=w1[is];
end;

{ # LES POINTS D'INTEGRATION POUR LES ELEMENTS # }
{ # 2-D ET 3-D TYPE 48,49, # }

```

```

procedure POINT1(var it,is:integer; var wt,ws,tt,ss:inter);
var
  s1,w1:inter;
begin
  wt[it]:=0 ; ws[is]:=0; tt[it]:=0; ss[is]:=0;
  s1[1]:=-0.7745966692414834;
  s1[2]:=0.0000000000000000;
  s1[3]:=-s1[1];
  w1[1]:=0.5555555555555555;
  w1[2]:=0.8888888888888888;
  w1[3]:=w1[1];
  tt[it]:=s1[it]+tt[it];
  ss[is]:=s1[is]+ss[is];
  wt[it]:=w1[it]+wt[it];
  ws[is]:=w1[is]+ws[is];
end;

```

```

{ # LES POINTS D'INTEGRATION POUR LES ELEMENTS # }
{ # 2-D ET 3-D DU TYPE 36, # }

```

```

procedure POINT36(var it:integer;var a136,b136,c136,wt36:inter);

```

```

var
    a17,b17,c17,w17:inter;

begin
    a136[it]:=0;b136[it]:=0;c136[it]:=0;wt36[it]:=0;
    a17[1]:=0.333333333333333333; a17[2]:=0.0597158717897698;
    a17[3]:=0.4701420641051151; a17[4]:=a17[3];
    a17[5]:=0.7974269853530873; a17[6]:=0.1012865073234563;
    a17[7]:=a17[6];
    b17[1]:=a17[1]; b17[2]:=a17[3]; b17[3]:=a17[2];
    b17[4]:=a17[3]; b17[5]:=a17[6]; b17[6]:=a17[5];
    b17[7]:=a17[6];
    c17[1]:=a17[6]; c17[2]:=a17[3]; c17[3]:=a17[3];
    c17[4]:=a17[2]; c17[5]:=a17[6]; c17[6]:=a17[6];
    c17[7]:=a17[5];
    w17[1]:=0.1125 ; w17[2]:=0.0629695902724136;
    w17[3]:=w17[2]; w17[4]:=w17[2];
    w17[5]:=0.0629695902724136; w17[6]:=w17[5];
    w17[7]:=w17[5];

    a136[it]:=a17[it]+a136[it];
    b136[it]:=b17[it]+b136[it];
    c136[it]:=c17[it]+c136[it];
    wt36[it]:=w17[it]+wt36[it];

end;

{ # ~~~~~# }
{ # CALCUL DE LA MATRICE DE RIGIDITE ET DU VECTEUR # }
{ # DE SOLlicitation ELEMENTAIRE POUR UN POINT DE # }
{ #          GUASS D'UN ELEMENT          # }
{ #                                     # }
{ ~~~~~ }

```

```

procedure GENEK(var xn,yn,zn:typcoord; var ke,perm:typemat; var w:real;

```

```

var qtm,fn,fnds,fndt,fndu:inter;var dimelt,nbnd:integer

var
  js,g,ing,jsg,bel,belt,bem,kp:typemat;
      deter:real;
      dims,l,k:integer;

begin
  dims:=3;
  deter:=0;
  for i:=1 to 3 do
  for j:=1 to 3 do
  begin
    js[i,j]:=0;
    jsg[i,j]:=0;
    g[i,j]:=0;
    ing[i,j]:=0;
  end;
  for i:=1 to 3 do
  for j:=1 to nbnd do
    bel[i,j]:=0;
  for i:=1 to nbnd do
  begin
    qtm[i]:=0;
    for j:=1 to nbnd do
    begin
      belt[i,j]:=0;
      bem[i,j]:=0;
      ke[i,j]:=0;
    end;
  end;
  for i:=1 to nbnd do
  for j:=1 to nbnd do
    kp[i,j]:=0;
  JACOBI(js,xn,yn,zn,fnds,fndt,fndu,nbnd);
  MATRIG(js,g,dimelt);

```

```

if dimelt=1
then
begin
deter:=g[1,1];
for i:=1 to nbnd do
jsg[i,1]:=(1/g[1,1])*js[i,1];
end
else
begin

INVERMATRI(g,ing,ind,dimelt,deter);
PRODAB(js,ing,jsg,dims,dimelt,dimelt);
end;
MATRIB(jsg,bel,fnds,fndt,fndu,dims,nbnd);
TRANSPO(bel,belt,dims,nbnd);
PRODAB(belt,perm,bem,nbnd,dims,dims);
PRODAB(bem,bel,kp,nbnd,nbnd,dims);

deter:=sqrt(deter);
for l:=1 to nbnd do
begin
qtm[l]:=fn[l]*(w*deter);
for k:=1 to nbnd do
begin
ke[l,k]:=deter*(w*kp[l,k]);
kp[l,k]:=0;
end;
end;
end;

end;
{ # CALCUL DE LA MATRICE DE RIGIDITE 'K ' ET DU VECTEUR # }
{ # DE SOLLICITATION 'FV ' DE L'ELEMENT TYPE-23 # }

```

Procedure K23(var xn,yn,zn:typecoord;var fek:inter;var perm,bb:typemat;nbnd

```

var
w,s:real;
qtm1,ss,ws:inter;
dimel : integer);

```

```

    ke,kl:typemat;
    i,j,is:integer;
begin
  for i:=1 to nbnd do
    begin
      fek[i]:=0;
      for j:=1 to nbnd do
        begin
          bb[i,j]:=0;
          ke[i,j]:=0;
          kl[i,j]:=0;
        end;
      end;
    for is:=1 to 3 do
      begin
        s:=0;
        w:=0;
        POINT23(is,ws,ss);
        s:=ss[is];
        w:=ws[is];
        for i:=1 to nbnd do
          begin
            fn[i]:=0;
            fnds[i]:=0;
            fndt[i]:=0;
            fndu[i]:=0;
          end;
        FN23(fn,fnds,s);
        for i:=1 to 3 do
          genek(xn,yn,zn,ke,perm,w,qtml,fn,fnds,fndt,fndu,dimelt,nbnd);
          for i:=1 to nbnd do
            begin
              fek[i]:=qtml[i]+fek[i];
              qtml[i]:=0;
            end;
          SOMATRI(ke,bb,kl,nbnd,nbnd);
          for i:=1 to nbnd do

```

```

        for j:=1 to nbnd do
            begin
                bb[i,j]:=kl[i,j];
                ke[i,j]:=0;
                kl[i,j]:=0;
            end;
        end;
end;
{ # CALCUL DE LA MATRICE DE RIGIDITE 'K ' ET DU VECTEUR # }
{ # DE SOLLICITATION 'FV 'ELEMENTAIRE DE L'ELEMENT      # }
{ #                TYPE-48,49                          # }

```

```

procedure K48(var xn,yn,zn:typecoord;var fek:inter;var perm,bb:typemat;nbnd:
var dimet: integer);

```

```

    w,s,t:real;
    qtml,ss,tt,wt,ws:inter;
    ke,kl:typemat;

    i,j,is,it:integer;

    begin
        for i:=1 to nbnd do
            begin
                fek[i]:=0;
                for j:=1 to nbnd do
                    begin
                        bb[i,j]:=0;
                        ke[i,j]:=0;
                        kl[i,j]:=0;
                    end;
                end;
                for it:=1 to 3 do
                    for is:=1 to 3 do
                        begin
                            s:=0; t:=0 ; w:=0;
                            POINT1(it,is,wt,ws,tt,ss);

```



```

s:=ss[is];
t:=tt[it];
w:=ws[is]*wt[it];
for i:=1 to nbnd do
begin
fn[i]:=0;
fnds[i]:=0;
fndt[i]:=0;
fndu[i]:=0;
end;
case nbnd of
8:FN48(fn,fnds,fndt,s,t);
9:FN49(fn,fnds,fndt,s,t);
end;
GENEK(xn,yn,zn,ke,perm,w,qtml,fn,fnds,fndt,fndu,dimelt,nbnd,
for i:=1 to nbnd do
begin
fek[i]:=qtml[i]+fek[i];
qtml[i]:=0;
end;
SOMATRI(ke,bb,kl,nbnd,nbnd);
for i:=1 to nbnd do
for j:=1 to nbnd do
begin
bb[i,j]:=kl[i,j];
ke[i,j]:=0;
kl[i,j]:=0;
end;
ss[is]:=0;tt[it]:=0;
end;
end;

{ # CALCUL DE LA MATRICE DE RIGIDITE ET DU VECTEUR # }
{ # DE SOLLICITATION ELEMENTAIRE DE L'ELEMENT # }
{ # TYPE-36 # }

```

```
procedure K36(var xn,yn,zn:typecoord;var fek:inter;var perm,bb:typemat;
              nbnd, dimelt:Integer);
```

```
var
ke,kl:typemat;
qtml,al36,b136,c136,wt36:inter;
    al,b1,c1,w:real;
    i,j,it:integer;
```

```
begin
```

```
  for i:=1 to nbnd do
```

```
    begin
```

```
      fek[i]:=0;
```

```
      for j:=1 to nbnd do
```

```
        begin
```

```
          bb[i,j]:=0;
```

```
          ke[i,j]:=0;
```

```
          kl[i,j]:=0;
```

```
        end;
```

```
      end;
```

```
    for it:=1 to 7 do
```

```
      begin
```

```
        al:=0; b1:=0; c1:=0; w:=0;
```

```
        POINT36(it,al36,b136,c136,wt36);
```

```
        al:=al36[it];
```

```
        b1:=b136[it];
```

```
        c1:=c136[it];
```

```
        w:=wt36[it];
```

```
        for i:=1 to nbnd do
```

```
          begin
```

```
            fn[i]:=0; fnds[i]:=0; fndt[i]:=0; fndu[i]:=0;
```

```
          end;
```

```
          FN36(fn,fnds,fndt,al,b1,c1);
```

```
          for i:=1 to nbnd do
```

```
            GENEK(xn,yn,zn,ke,perm,w,qtml,fn,fnds,fndt,fndu,dimelt,
```

```
            for i:=1 to nbnd do
```

```
              begin
```

```
                nbnd);
```

```

    fek[i]:=fek[i]+qtm1[i];
    qtm1[i]:=0;
    end;
    SOMATRI(ke,bb,kl,nbnd,nbnd);
    for i:=1 to nbnd do
        for j:=1 to nbnd do
            begin
                bb[i,j]:=kl[i,j];
                ke[i,j]:=0;
                kl[i,j]:=0;
            end;
            al36[it]:=0; bl36[it]:=0; cl36[it]:=0; wt36[it]:=0;
        end;
    end;
end;

```

```

{ #~~~~~# }
{ # ASSEMBLAGE DE LA MATRICE ELEMENTAIRE 'GEK ' ET # }
{ # DU VECTEUR ELEMENTAIRE DE SOLLICITATOIN 'FEK ' # }
{ # DANS LA MATRICE DE GLOBALE 'GEG ' ET LE VECTEUR # }
{ # DE SOLLICITATION GLOBAL 'FEG 'SUIVANT LA METHODE # }
{ # FRONTALE (triagulation et elimination eventuel) # }
{ # # }
{ ~~~~~ }

```

```

Procedure ASSEMBLAGE(Var vek,cek:inter1; var fek :inter ;var gek:typemat;
                    var dimcek:integer);

```

```

Var
    c,l,fac,r:integer;
    npert,int1,int2:real;
    nvp,nvpt:integer;

```

```

BEGIN
    nvp:=0;
    for i:=1 to dimcek do
        begin
            if cek[i]<=0

```

```

    THEN
        begin
            r:=1;
            While vek[i]<>veg[r] do
                r:=r+1;
                vek[i]:=r;
            end
        ELSE
            begin
                r:=1;
                While veg[r]<>0 do
                    r:=r+1;
                    veg[r]:=vek[i];
                    vek[i]:=r;
                    nvp:=nvp+1;
                end;
            end;
        end;

        { # assemblage # }

for i:=1 to dimcek do
for j:=1 to dimcek do
    geg[vek[i],vek[j]]:=geg[vek[i],vek[j]]+gek[i,j];
for i:=1 to dimcek do
    begin
        fegd[vek[i]]:=fegd[vek[i]]+fek[i];
        feg[vek[i]]:=feg[vek[i]]+fek[i];
    end;
nvpt:=nvp-pel;
If nvpt>=0
    then
        dimb:=dimb+nvpt
    else
        dimb:=dimb+0;
        pel:=0;

i:=1;

```

```

repeat
  if (cek[i]=0) or (cek[i]=1)
    THEN
      begin
        if hhexx[i]=-1
          then
            { # triangulation non standart pour des charges connues # }
            begin
              for l:=1 to dimb do
                begin
                  if l=vek[i]
                    then
                      fac:=0
                    else
                      fac:=1;
                  feg[l]:=feg[l]-(geg[l,vek[i]]*valhh[i]*fac);
                  geg[l,vek[i]]:=geg[l,vek[i]]*(1-fac);
                end;
              end
            else
              { # triangulation standart pour les charges inconnues # }
              begin
                if abs(geg[vek[i],vek[i]]) < 1.E-10
                  then
                    begin
                      clrscr;
                      writeln('          PROBLEME NON DEFINIT');
                      writeln('      Condition aux limites insuffisant ');
                      writeln('          Revoir les entres');
                      delay(4000);
                      halt;
                    end;
                for l:=1 to dimb do
                  begin
                    if l=vek[i]
                      then
                        fac:=0

```

```

        else
            fac:=1;
        npert:=geg[l,vek[i]];
        for c:=1 to dimb do
            begin
                int1:=((npert*geg[vek[i],c])/(geg[vek[i],vek[i]]))*fac
                geg[l,c]:=geg[l,c]-int1 ;
            end;
            int2:=((npert*feg[vek[i]])/(geg[vek[i],vek[i]]))*fac ;
            feg[l]:=feg[l]-int2;
        end;
    end;

    with equax do

{ # enregistrement et elimination de la ligne # }
{ # correspondante a la variable qui apparait # }
{ #           pour la dernier fois           # }

        begin
            for c:=1 to dimb do
                begin
                    coefx[c]:=geg[vek[i],c];
                    geg[vek[i],c]:=0;
                end;
            coeff:=feg[vek[i]];
            feg[vek[i]]:=0;
            place:=vek[i];
            numero:=veg[vek[i]];
            veg[vek[i]]:=0;
            long:=dimb;
            cxx:=hhcxx[i];
            lhh:=valhh[i];
            debp:=fegd[vek[i]];
            fegd[vek[i]]:=0;
            e:=e+1;
            ENREG(equax);
        end;
    end;

```

```

        end;
        pel:=pel+1;
        i:=i+1;
    end
ELSE
    i:=i+1;
UNTIL I>dimcek;
for l:=1 to dimcek do
for c:=1 to dimcek do
    gek[l,c]:=0;
for l:=1 to dimcek do
    begin
        vek[l]:=0;fek[l]:=0;cek[l]:=0;
    end;
END;

    { # RESOLUTION DU SYSTEME GLOBAL PAR LA # }
    { # METHODE DE SUBSTITUTION INVERSE # }

```

Procedure RESOLUTION(var nbtnd:integer);

```

VAR
    val:inter;
    l:integer;
    int3:real;
BEGIN
    for i:=1 to 20 do
    for j:=1 to 20 do
        geg[i,j]:=0;
    for i:=1 to 20 do
        begin
            val[i]:=0;
            veg[i]:=0;
        end;
    np:=nbtnd;
    REPEAT
        with equax do
        with valsol do

```

```

begin
  vari:=0;
  solh:=0;
  solq:=0;
  coeff:=0;
  place:=0; debp:=0;
  numero:=0;
  long:=0;
  for i:=1 to dimb do
    coefx[i]:=0;
  RECHERCHE(equax,Np);
  if veg[place]<>0
    Then
      begin
        val[place]:=0;
        veg[place]:=numero;
      end

    else
      veg[place]:=numero;

    int3:=0;
    for l:=1 to long do
      int3:=int3+coefx[l]*val[l];
    if cxx=-1
      then
{ # calcul du debit quand la charge est connue # }
      begin
        coeff:=debp+(coefx[place]*lhh)+int3-coeff;
        val[place]:=lhh;
      end
    else
{ # calcul de la charge quand le debit est connu # }
      begin
        val[place]:=(coeff-int3)/(coefx[place]);
        coeff:=debp;
      end;

```



```

        if coeff < 0
            then
                somdbneg:=somdbneg + coeff
            else
                somdbpos:=somdbpos + coeff;

        vari:=veg[place];
        solh:=val[place];
        solq:=coeff;
        RESULTAT(valsol);
        np:=np-1;
    end;
UNTIL np<1;
total:=somdbneg+somdbpos;
END;

```

PROCEDURE PRESENT;

```

begin
    GotoXY(6,5);
    Writeln('~~~~~');
    writeln(' |
    writeln(' |
    gotoxy(4,whereY);
    writeln(' |          CE LOGICIEL PERMET DE SIMULER PAR LA METHODE DES
    gotoxy(4,wherey);
    writeln(' |          ELEMEMENTS FINIS L'ECOULEMENT PERMANENT EN
    gotoxy(4,wherey);
    writeln(' |          MILIEU POREUX SATURE
    gotoxy(4,wherey);
    writeln(' | Il utilise des elements 1-D,2-D et 3-D dans un espace 3-D
    writeln(' |

```

```

writeln('      ;
gotoxy(6,wherey);
writeln('~~~~~
delay(4000);
clrscr;
end;

```

```

PROCEDURE OPT1;
begin
  gotoxy(4,4);
  writeln('                OPTION-1
  writeln;
  writeln('          1 - Creation de nouvelles donnees
  writeln;
  writeln('          2 - Modification ou Creation de certains fichiers
  writeln;
  writeln('          3 - Execution
  writeln;
  writeln('                Faites votre choix
  gotoxy(13,13);
  readln(option1);
  clrscr;
end;

```

```

PROCEDURE OPT2;
BEGIN
  gotoxy(4,4);
  writeln('                OPTION-2
  writeln;
  writeln('          21 - Fichier des classes de permeabilite
  writeln;
  writeln('          22 - Fichier des connectivites
  writeln;
  writeln('          23 - Fichier des coordonnees
  writeln;

```

```

writeln('                Faites votre choix
gotoXY(13,13);
readln(option2);
clrscr;
gotoxy(4,5);
writeln('                OPTION-3
writeln;
writeln('          31 - Modification
writeln;
writeln('          32 - Creation
GotoXY(13,13);
readln(option3);

end;

PROCEDURE OPT4;
begin
  clrscr;
  gotoxy(4,6);
  writeln('                OPTION-4
  writeln;
  writeln('          41 - impression des resultats sur ecran ');
  writeln;
  writeln('          42 - Impression sur imprimante
  GotoXY(13,13);
  readln(option4);
  clrscr;
end;

BEGIN
  Clrscr;

```

```

DEBUT;
PRESENT;
OPT1;
While option1 <> 3 do
  Begin
    Repeat
      if option1 = 2
        then
          OPT2;
          SAISIE_ET_CONTROLE_DE_DONNEES;
          Clrscr;
          OPT1;
        Until option1=3;
    end;
    OPT4;
    gotoxy(5,5);
write('          EN EXECUTION          ');
sombdbneg:=0;
sombdbpos:=0;
total:=0;
For i:=1 to 20 do
begin
feg[i]:=0;
veg[i]:=0;
fegd[i]:=0;
For j:=1 to 20 do
  geg[i,j]:=0;
end;
e:=0;
dimb:=0;
pel:=0;
VECTIDI;
ASSIGN(loce, 'B:NENO.DAT');
ASSIGN(corg, 'B:NENO.COR');
ASSIGN(fiche_alpha, 'B:NENOPERM.DAT');
RESET(loce);
RESET(corg);

```

```

RESET(fiche_alpha);
WITH Elt do
WITH coord do
WITH permeabilite do
  begin
    For kp:=1 to nbtelt do
      begin
        SEEK(loce,kp-1);
        READ(loce,Elt);
        qs:=ald;
        CHOIX2;
        For i:=1 to nbnd do
          begin
            SEEK(corg,noeud[i]-1);
            READ(corg,coord);
            xn[i]:=xxn[1];
            yn[i]:=xxn[2];
            zn[i]:=xxn[3];
            hhcx[i]:=hhcx;
            valhh[i]:=valh;
            valqsp[i]:=valqsp;
          end;
        SEEK(fiche_alpha,classe-1);
        READ(fiche_alpha,permeabilite);
        perm[1,1]:=K[1];
        perm[1,2]:=K[2];
        perm[1,3]:=K[3];
        perm[2,1]:=K[2];
        perm[2,2]:=K[4];
        perm[2,3]:=K[5];
        perm[3,1]:=K[3];
        perm[3,2]:=K[5];
        perm[3,3]:=K[6];
        for i:=1 to 10 do
          fek[i]:=0;
        Case typelmt of
          23 : K23(xn,yn,zn,fek,perm,bb,nbnd,dimelt);

```

```

36 : k36(xn,yn,zn,fek,perm,bb,nbnd,dimelt);
48 : k48(xn,yn,zn,fek,perm,bb,nbnd,dimelt);
49 : K48(xn,yn,zn,fek,perm,bb,nbnd,dimelt);
end;
For i:=1 to nbnd do
  fek[i]:=qs*fek[i]+(valqspp[i]/idi[i]);
For i:=1 to 10 do
  begin
    vek[i]:=0;
    cek[i]:=0;
  end;
  VECVEK;
  VECTCEK;
  ASSEMBLAGE(vek,cek,fek,bb,nbnd);
end;
end;
CLOSE(loce);
CLOSE(corg);
CLOSE(fiche_alpha);
RESOLUTION(nbnd);
Clrscr;
if option4 = 41
  then
    IMPRESSION2
  else
    IMPRESSION1 ;
END.

```

PROCEDURE SAISIE_ET_CONTROLE_DE_DONNEES;

VAR i,j,k,l,direct,max:integer;

Lettre:char;Posi:Array[1..8] of integer;

posis,Posit:array[1..30] of integer;

PROCEDURE SAISIE3;

BEGIN

ASSIGN(Fiche_alpha,'b:NENOPERM.DAT');

Rewrite(Fiche_alpha);

With Permeabilite Do

Begin

clrscr;

Write(' DONNER LE NOMBRE DE CLASSE DE FERMEABILITE : ');

read(nbreperm);

Clrscr;

Write(' SAISIE DE ',nbreperm,' CLASSES DE FERMEABILITE ');

FOR I:=1 TO 4 DO WRITELN;

Write('Nx ');Posi[1]:=WhereX+1;

Write('CLASSE ');Posi[2]:=WhereX+1;

Write('K1 ');Posi[3]:=WhereX+1;

Write('K2 ');Posi[4]:=WhereX+1;

Write('K3 ');Posi[5]:=WhereX+1;

Write('K4 ');Posi[6]:=WhereX+1;

Write('K5 ');Posi[7]:=WhereX+1;

Write('K6');Posi[8]:=WhereX+1;Writeln;

FOR I:=1 TO 80 DO WRITE('-');

Writeln;

```

FOR I:=1 TO Nbreperm DO
    Begin
        Write( I );GOTOXY(Posi[1],WhereY);
        Read(Classperm);
        GOTOXY(Posi[2],WhereY);
        For j:=1 to 6 do
            Begin
                Read(K[j]);
                GotoXY(posi[J+2],WhereY);
            End;
        Write(Fiche_alpha,permeabilite);
        Writeln;
    End;
End;
Writeln;
Writeln;
Flush(fiche_alpha);
Close(fiche_alpha);
Write('Entrer O(ui) pour controle sinon N(on) ');
Readln(Lettre);
IF (lettre='O') or (lettre='o') Then
    Begin
        Assign(fiche_alpha,'b:NENOPERM.DAT');
        Reset(Fiche_alpha);
        clrscr;
Write('          CONTROLE DE PERMEABILITE          NOMBRE....',Nbreperm)
        Writeln;
        writeln;
    End;

```



```

for i:=1 to nbreperm do
Begin
  SEEK(fiche_alpha,i-1);
  Read(fiche_alpha,permeabilite);
  With Permeabilite do
    Begin
      Writeln('AN Nx CLASSE      K1      K2      K3      K4      K5      K6
      Write('      ',I);
      gotoXY(Posi[1]+2,WhereY+1);
      Write(classperm);
      for j:=1 to 6 do
        begin
          gotoXY(posi[J+1]+2,WhereY);
          Write(K[J]:5:3);
        End;
      writeln;
      Write('NV',I);
      gotoxy(posi[1]+2,whereY);
      Read(classperm);
      For j:=1 to 6 do
        begin
          gotoXY(posi[J+1]+2,whereY);
          Read(K[J]);
        End;
      clrscr;
    end;
  SEEK(fiche_alpha,i-1);

```

```
Write(fiche_alpha,permeabilite);
```

```
End;
```

```
Flush(fiche_alpha);
```

```
Close(fiche_alpha);
```

```
end;
```

```
End;
```

```
PROCEDURE CHOIX;
```

```
Begin
```

```
    CASE Elt.typelmt OF
```

```
        23: L:=3;
```

```
        36: L:=6;
```

```
        48: L:=8;
```

```
        49: L:=9;
```

```
        410: L:=10;
```

```
    End;
```

```
End;
```

```
PROCEDURE SAISIE1;
```

```
Begin
```

```
    Clrscr;
```

```
    write(' DONNER LE NOMBRE D'ELEMENTS      : ');
```

```
    Read(nbtelt);
```

```
    Clrscr;
```

```
    Writeln('SAISIE DES DONNEES      ',nbtelt,' ELEMENTS');
```

```
    Writeln;Writeln;
```

```

Assign(Loce,'b:nenodat');
Rewrite(Loce);
With elt do
  Begin
    Write('Element  ');posit[1]:=whereX+1;
    Write('Tpel  ');posit[2]:=whereX+1;
    Write('Classe ');posit[3]:=whereX+1;
    Write('ALD  ');posit[4]:=whereX+1;
    For I:=1 to 10 do
      begin
        Write('Nd',I,' ');
        posit[I+4]:=whereX+1;
      end;
    Writeln;
    for i:=1 to 80 do
      Write('_');
      For I:=1 to nbtelt do
        Begin
          Write('nx',I,' ');gotoXY(posit[1],whereY);
          Read(Typelmt);
          GotoXY(posit[2],whereY);
          Read(Classe);
          gotoXY(posit[3],whereY);
          Read(Ald);
          GotoXY(posit[4],whereY);
          Choix;
          For K:=1 to L do
            Begin

```

```

        Read(Noeud[K]);
        GotoXY(posit[K+4],WhereY);
    End;
    Writeln;
    Write(Loce,Elt);
End;
Writeln;writeln; Flush(Loce);Close(Loce);
Writeln('SAISIE TERMINEE ');
Write('Entrer o(ui) pour le controle n(on pour sortir ');
Readln(Lettre);
While (Lettre<>'O') and (Lettre<>'o') and (Lettre<>'N') and
    Begin
        Write('Lettre invalide entrer une autre ');
        Readln(Lettre);
    End;
If (Lettre='O') or (Lettre='o') Then
    Begin
        Assign(Loce,'b:Neno.dat');
        Reset(Loce);
        Clrscr;
        Writeln('CONTROLE DES DONNEES ');
        Writeln;writeln;
        for I:=1 to nbtelt do
            Begin
                Clrscr;
                SEEK(loce,i-1);
                Read(Loce,Elt);
                Writeln('Element nx',I);
            End;
        End;

```

```

        Writeln('*****');
        Write('Type element: ', Typelmt, ' Modification:
        Readln(Typelmt);
        Write('Classe de permeabilite: ', Classe, ' Modif
        Readln(Classe);
        Write('Alimentation distribue: ', Ald:5:3, ' Modif
        Readln(ald);
        Choix;
        For K:=1 to L do
            Begin
                Write('
                Readln(Noeud[K]);
            End;
        seek(loce, i-1);
        Write(Loce, Elt);
    End;
End;

```

```

Flush(Loce);
Close(Loce);
End;

```

End;

PROCEDURE SAISIE2;

```

Begin
    Assign(Corg, 'E: neno.cor');
    Rewrite(Corg);
    With Coord do
        Begin

```

```

                Writeln('*****');
Write('Type element: ',Typelmt,' Modification: ');
Readln(Typelmt);
Write('Classe de permeabilite: ',Classe,' Modification:');
Readln(Classe);
Write('Alimentation distribuë:',Ald:5:3,' Modification');
Readln(ald);
    Choix;
        For K:=1 to L do
            Begin
Write('                Noeud',K,':',Noeud[K],' Modification :');
            Readln(Noeud[K]);
                End;
                seek(loce,i-1);
                Write(Loce,Elt);
            End;
        End;

        Flush(Loce);
        Close(Loce);
        End;
End;

```

PROCEDURE SAISIE2;

```

Begin
    Assign(Corg,'B:nenocor');
    Rewrite(Corg);
    With Coord do
        Begin

```

```

    clrscr;
    write( '      DONNER LE NOMBRE TOTAL DE NOEUDS      :');
    read(nbtnd);
    Clrscr;
Writeln('SAISIE DES COORDONNEES      ',Nbtnd,' NOEUDS');
    Writeln;Writeln;Writeln;
    Write('Noeud nx  ');posis[1]:=whereX+1;
    write('Xcoord  ');posis[2]:=whereX+1;
    write('Ycoord  ');posis[3]:=whereX+1;
    write('Zcoord  ');posis[4]:=whereX+1;
    write(' Hhcx   ');posis[5]:=whereX+1;
    write(' Valh   ');posis[6]:=whereX+1;
    write(' valqsp ');posis[7]:=whereX+1;
    writeln;
    for i:=1 to 80 do
    write('_');
    writeln;
    For I:=1 to Nbtnd do
        Begin
            Write(I);gotoXY(posis[1],whereY);
            For K:=1 to 3 do
                begin
                    Read(xxn[K]);
                    gotoXY(posis[K+1],whereY);
                end;
            Read(Hhcx);gotoXY(posis[5],WhereY);
            Read(Valh);gotoXY(posis[6],WhereY);
            Read(valqsp);gotoXY(posis[7],whereY);

```

```

Write(Corg,coord);
writeln;
If I MOD 18 =0 Then
  Begin
    Clrscr;
    Writeln('Noeud nx  ,Coordonnee Hhcx Valh');
  End;
End;
Writeln;Writeln;
Writeln('SAISIE DE COORDONNEES  TERMINEE');
Writeln('Entrer o(ui pour le controle ,n(on pour sortir ');
Readln(Lettre)
While (Lettre<>'N') and (Lettre<>'n') and (Lettre<>'o') and (Lettre<>'O') c
  Begin
    Write('Lettre invalide entrer une autre ');
    Readln(Lettre);
  End;
If (Lettre='O') or (Lettre='o') Then
Begin
Clrscr;
Writeln('Noeud nx  Coordonnees Hhcx Valh');
for I:=1 to nbtd do
  Begin
    SEEK(corg,i-1);
    Read(Corg,coord);
    Write(I:4,' ( ');
    For K:=1 to 3 do
      Begin

```



```

        Write(xxn[K]:5:3);
        If K<3 Then Write(' ', ' ');
    End;
Write(' ) ');Write(Hhcx, ' ',valh:5:3,' ',valqsp);
Write(' Mod ( ');
For K:=1 to 3 do
    Begin
        Read(Xn[K]);
        If K<3 Then Write(' ', ' ');
    End;
Write(' ) ');Read(Hhcx);write(' ');
Read(Valh);write(' ');
read(valqsp); Writeln;
SEEK(corg,i-1);
Write(Corg,Coord);
If I MOD 18 =0 Then
    Begin
        Clrscr;
        Writeln('Noeud nx   Coordonnee  Hhcx  Valh');
    End;
End;
End;
Writeln;Writeln;
Writeln('CONTROLE TERMINE');
End; Flush(Corg);Close(Corg);
End;
BEGIN
Case option1 of

```

```
1: begin
    SAISIE3;
    SAISIE1;
    SAISIE2;
end;
2: begin
    Case option2 of
        21:SAISIE3;
        22:SAISIE1;
        23:SAISIE2;
    end;;
end;
end;
END;
```

BIBLIOGRAPHIE

- 1 - GOURI DHATT et TOUZOT : Une présentation de la méthode des éléments finis.
Les presses de l'université larat 1981.
- 2 - O.C. ZIENKIEWICZ : La méthode des éléments finis appliquée à l'art de l'ingénieur . Paris . 1973
- 3 - J. TINSLEY ODEN et GRAHAM F. CAREY
Finite éléments
 - 1 - Introduction
 - 2 - A second course
 - 3 - Computational aspects
 - 4 - Mathematical aspects
 - 5 - Special problems in solid mechanics
fluid mechanicsDans la collection the Texas finite element series.
- 4 - G. de MARSILY : Hydrogéologie qualitative . MASSON . 1981

5. HENRI VARCOLLIER : exposés et exemples
d'applications de calcul tensoriel et de
calcul matriciel. à l'usage des Ingé-
nieurs et physiciens.

Troisième édition. Presses universitaires de
France

6. Bathe K.J and Wilson E.L. Numerical me-
thods in finite element analysis,
Prentice-Hall, Englewood Cliffs, N.J., 1976

7. Peter GREGOND : La programmation en
Pascal. ADDISON-WELLEY EUROPE.

8. Technical report : par KIRALY 1979
du centre de recherche CEDRA (Suisse)

9. G. CASPARY Traité pratique des
eaux souterraines Paris 1967