

REPUBLIQUE DU SENEGAL

ECOLE POLYTECHNIQUE DE THIES

GC 0658

PROJET DE FIN D'ETUDE
en vue de l'obtention du diplôme d'ingénieur de conception

TITRE ANALYSE DES CONTRAINTES DANS LES SYSTEMES
MULTICOUCHES

DATE: JUILLET 1991

AUTEUR: Victor ZOUNMENO
DIRECTEUR: Massamba DIENE
CO-DIRECTEUR

DEDICACE

A mes chers parents: Papa, maman et frères, pour qu'ils y trouvent, le fruit de tant d'années de peines, qu'ils ont consenties pour la réussite de mon éducation.

A tous mes amis, qui ont grandement contribué à la réussite de ma formation.

Victor ZOUNMENO

REMERCIEMENTS

Le présent projet ne peut être achevé sans le témoignage de nos remerciements à tous ceux qui, de près ou de loin, ont contribué à sa mise au point.

Qu'il nous soit permis d'exprimer toute notre gratitude envers M^r **Massamba DIENE**, professeur à l'Ecole Polytechnique de Thiès, département Génie Civil, qui malgré toutes ses préoccupations, a su diriger, avec une attention particulière, notre travail.

Nous tenons à remercier tous les professeurs du Tronc Commun et du département Génie Civil pour la formation qu'ils su nous donner.

Nous adressons notre reconnaissance à tout le presonnel pédagogique pour leur apport qui a été d'un grand secours.

Merci à tous ceux qui, d'une manière ou d'une autre, ont contribué à l'achèvement de ce projet.

SOMMAIRE

L'objet essentiel de ce travail est l'analyse des contraintes dans les systèmes multicouches, cas de route. Cette analyse fait suite à une partie théorique qui est plus consacrée à la description des modèles de calcul de contraintes existants.

Le chapitre II est consacré à la description des structures de chaussées. Le chapitre III qui aborde la description des modèles de calcul de contraintes fait cerner, à travers des formulations mathématiques, la distribution des contraintes sous une charge circulaire appliquée à la surface d'un massif granulaire. Alors que dans le chapitre IV il est question du choix d'objectif, dans la conception d'une structure routière.

Enfin, dans le chapitre V, nous appliquons la théorie isotrope de Burmister, au calcul de contraintes et au dimensionnement d'une superstructure routière.

TABLE DES MATIERES

TITRES	PAGES
DEDICACE	i
REMERCIEMENTS	ii
SOMMAIRE	iii
TABLE DES MATIERES	iv
LISTE DES FIGURES	v
LISTE DES ANNEXES	vi
SYMBOLES UTILISÉS	vii
Chapitre I : INTRODUCTION	1
Chapitre II : STRUCTURES DE CHAUSSEES	4
2.1) Structure souple	4
2.2) Structure rigide	6
Chapitre III: APERÇU SUR LES MODELES DE CALCULS DE CONTRAINTES EXISTANTS	8
3.1) Paramètres de calcul	8
3.1.1_ Environnement	8
3.1.2_ Modes de Chargement	10
3.1.3_ Trafic	11
3.2) Modèles de la mécanique des chaussées	12
3.2.1_ Modèle de BOUSSINESQ	12
3.2.2_ Modèle de HOGG	14
3.2.3_ Modèle de WESTERGARD	15
3.2.4_ Modèle de BURMISTER	17

3.2.5_	Modèle tenant compte de l'anisotropie	23
3.2.6_	Modèle d'éléments finis	26
3.3)	Confrontation des modèles	29
Chapitre IV	DIMENSIONNEMENT DES SUPERSTRUCTURES ROUTIERES.	30
4.1)	Concept de sécurité et critères de rupture	30
4.2)	Calcul des épaisseurs	32
4.3)	Lois de fatigue	33
4.3.1_	Rupture en fatigue par flexion répétée des matériaux traités aux liants hydrauliques ..	33
4.3.2_	Matériaux bitumineux	33
4.3.2.1	Rupture par fissuration de l'enrobé	33
4.3.2.2	Orniérage	34
4.3.3_	Matériaux non traités	37
4.4)	Détermination de la durée de vie de la superstructure routière	38
Chapitre V	PROGRAMMATION AVEC LE MODELE DE BURMISTER UTILISANT L'ELASTICITE ISOTROPE	40
5.1)	Raisons du choix du modèle	40
5.2)	Différentes étapes du programme	41
5.2.1_	L'établissement du système d'équations linéaires	41
5.2.2_	Calculs d'intégrales	44
5.3)	Caractéristiques du logiciel Zovroute	46
5.3.1_	Contraintes sous roues jumelées	52
5.3.2_	Interprétation des résultats	52
5.4)	Définition des touches	56
5.5)	Limites du programme	57
Chapitre VI	CONCLUSION ET RECOMMANDATIONS	59
	ANNEXES	61

A₁ - Courbes et racines des fonctions de Bessel J_0 et J_1 .	62
A₂ - Organigramme	65
A₃ - Listing du programme constituant le logiciel	68

LISTE DES FIGURES

		Pages
Figure 2.1	Coupe d'une structure souple	6
Figure 2.3	Coupe d'une structure rigide	6
Figure 2.4	Coupe d'une structure type	7
Figure 3.1	Chargement de la chaussée	10
Figure 3.2	Contraintes sur un élément du massif	13
Figure 3.3	Modèle de Hogg	15
Figure 3.4	Modèle de Westergaard	16
Figure 3.5	Modèle de Burmister	17
Figure 3.6	Modèle d'éléments finis	26
Figure 4.1	Variation du déviateur en fonction de la température	35

LISTE DES ANNEXES

- A1. Courbes et racines des fonctions de Bessel J_0 et J_1 dans l'intervalle $[0,30]$
- A2. Organigramme du Programme
- A3. Listing du Programme et Résultats.

SYMBOLES UTILISES

- [] : Indication de référence
- a : Rayon de l'empreinte du pneumatique
- z : Profondeur à partir de la surface du sol
- h : Epaisseur d'une couche
- r : Excentricité par rapport à l'axe du cercle de chargement
- μ : Coefficient de Poisson
- $\rho = r/a$: Rayon adimensionnel
- E_0 : Intégrale elliptique de seconde espèce
- π_0 : Intégrale elliptique de troisième espèce
- E : Modules élastiques
- σ : Contrainte normale
- τ : Cisaillement
- ϵ : Déformation linéaire
- ϕ : Fonction de contraintes
- δ : Déformation angulaire
- u : Déplacement horizontal
- w : Déplacement vertical ou déflexion
- J_n : Fonction de Bessel de première espèce d'ordre n
- n : Nombre de couches
- $\hat{\Delta}$: Laplacien

INTRODUCTION

Le développement démographique et économique du XIX^e siècle ne serait pas pensable sans la réalisation d'un réseau routier assurant la circulation et le transport de marchandises entre tous les points du territoire. L'importance d'un tel réseau devient apparente, lors des travaux d'entretien, qui engendrent la coupure de certains tronçons.

Les problèmes que posent la conception, la construction et l'exploitation des ouvrages routiers procèdent des mêmes démarches que ceux posés par toutes les infrastructures de transport (aéroports, ports maritimes, voies ferrées etc.).

Etant la plus ancienne des infrastructures de transport la route a subi une évolution dans sa conception et sa construction, avec le temps, sous l'effet du trafic et des intempéries existantes. C'est ainsi que des routes monocouches on a abouti à des routes en système multicouche avec une surface de roulement en terre, à base de liants hydrauliques ou à base de liants hydrocarbonés. Les méthodes empiriques utilisées par le passé pour le dimensionnement et la construction des routes sont actuellement de plus en plus délaissées au profit des méthodes scientifiques plus élaborées. Ces méthodes ont connu une amélioration avec l'avancement des recherches sur les propriétés des matériaux de

construction.

Dans les pays en voie de développement, comme nos pays d'Afrique tropicale, la réalisation d'un réseau routier économique et fiable est une condition impérative pour assurer un développement durable.

De nos jours, avec la croissance du trafic la construction des routes nécessite la mise en place de matériaux pierreux de bonne qualité et souvent, l'utilisation de liants tels que le bitume et le ciment dont la fabrication, le transport, le traitement et la mise en place nécessitent une consommation d'énergie importante. De plus, des difficultés toujours de plus en plus grandes, rencontrées dans l'approvisionnement en matériaux d'une part et la pénurie croissante de sources d'énergie d'autre part, contraignent l'ingénieur à utiliser des méthodes de dimensionnement et des modes de construction permettant une utilisation optimale des ressources en fonction du trafic.

Dans cette perspective, l'utilisation des matériaux locaux, dans la construction routière, constitue un moyen privilégié pour réduire un tant soit peu les dépenses de nos Etats démunis face aux urgences financières quotidiennes.

Le dimensionnement de la chaussée qui est en fait la conception de sa structure est une étape importante dans la construction routière. Pour réaliser cette étape, il est nécessaire de déterminer les efforts transmis au sol par le biais de la structure de chaussée par des agents extérieurs comme le trafic. Ainsi bon nombre d'auteurs ont établi des modèles de calcul de contraintes dans la structure de la chaussée. Ces modèles ont évolué dans le temps suite à des recherches effectuées sur le

comportement rhéologique des matériaux de chaussée.

Le modèle le plus utilisé de nos jours est celui de Burmister [2].

C'est sur ce modèle que par exemple le programme du LCPC ALIZE s'est basé pour établir les abaques ou les catalogues qu'utilisent bon nombre de pays africains, surtout les pays tropicaux pour dimensionner les chaussées [1].

Dans le cadre de ce projet nous nous proposons d'adapter ce modèle à l'informatique .

Il s'agit d'élaborer un programme de calcul des contraintes dans les différentes couches de la superstructure routière, tenant compte des propriétés d'élasticité des matériaux éventuellement utilisés.

STRUCTURES DE CHAUSSEES

La structure de chaussée est un empilement de matériaux plus ou moins épais disposé en une ou plusieurs couches bien individualisées ayant chacune un rôle déterminé.

Elle a pour fonction de supporter des charges mobiles que l'on schématise conventionnellement par des forces réparties sur une surface circulaire, de protéger le sol de la plateforme qui est de faible capacité portante et parfois très sensible à l'eau.

Selon le mode transmission des charges on peut avoir:

2.1 Structures souples.

On assiste à une transmission prononcée de contraintes et à une déformation plus ou moins importante au passage des charges roulantes.

Dans le but d'épanouir les charges maximales parvenant au sol-support on doit avoir des couches de qualité géotechnique améliorée de bas en haut. On distingue:

Sol de plateforme.

C'est le sol en place lorsque la route est en déblai ou le sol d'emprunt lorsqu'elle est en remblai. Généralement il est traité lorsqu'il est très compressible. Son épaisseur est considérée infinie (assez grande).

_ Couche de forme.

Elle est pour pallier l'insuffisance du sol naturel. Elle permet d'améliorer la portance du sol-support et facilite le compactage des couches supérieures en leur fournissant un support ferme.

_ Couche de fondation.

Son rôle principal est de réduire les charges qui sont transmises à la plateforme. Elle doit être peu déformable et plus résistante que les couches sous-jacentes.

Généralement on rend sa partie inférieure peu perméable afin d'éviter l'infiltration à travers la couche de base. Cette partie empêche aussi la remontée capillaire des eaux de la plateforme.

_ Couche de base.

A ce niveau, les efforts dus au trafic sont encore très importants de même que les effets des facteurs environnementaux. Et les matériaux utilisés doivent donc être de bonne qualité géotechnique c'est-à-dire:

- . avoir un angle de frottement interne satisfaisant,
- . contenir moins de fines.

Son rôle principal est d'augmenter la capacité portante de la structure et de résister à l'érosion de toute nature.

_ Couche de revêtement.

C'est un mélange d'agrégats de bonne qualité et de liant hydrocarboné. Le revêtement doit avoir une bonne résistance au poinçonnement et à l'usure puisqu'il est en contact direct avec l'atmosphère et les sollicitations, avoir un bon uni et être étanche et peu glissant.

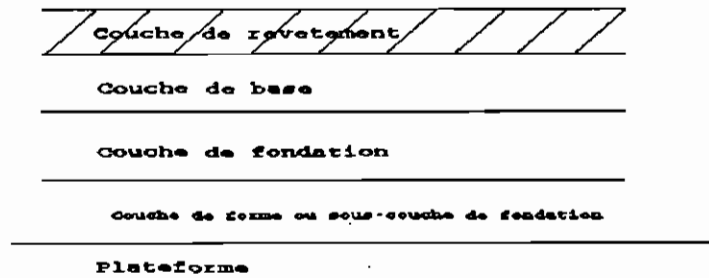


Fig 2.1 Structure souple

2.2 Structures rigides

Elles sont constituées d'une dalle reposant, soit directement sur le sol naturel soit par l'intermédiaire d'une couche de fondation.

Le sol naturel, lorsqu'il est appelé à supporter la dalle ne doit pas être source de gonflement ou de tassements différentiels.

La fondation, en plus de sa fonction d'augmenter la portance de la structure, doit fournir à la dalle une surface portante uniforme et remplacer les sols mous très compressibles.

Le revêtement qui est généralement une dalle de béton transforme la majeure partie des contraintes en contraintes de traction vers la base de la dalle.

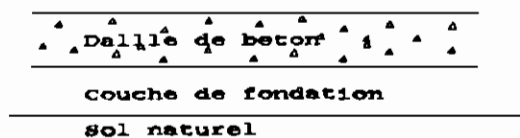


Fig 2.3 Coupe d'une structure rigide.

Il y a aussi des structures-hybrides qui sont soit des chaussées en béton très peu dosé avec de revêtements hydrocarbonés soit des chaussées souples avec certaines couches stabilisées au ciment.

Mais la structure type peut être représentée comme suit:

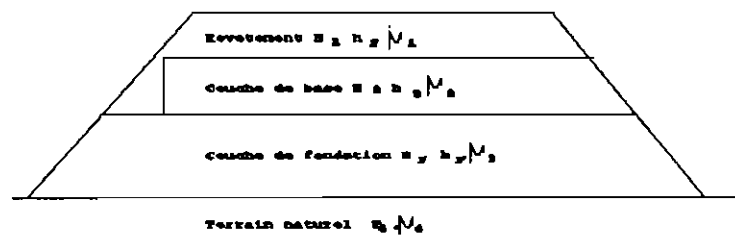


Fig 2.4 Coupe d'une structure type.

**APERCU SUR LES MODELES EXISTANTS
DE CALCUL DES CONTRAINTES**

Les modèles existants se basent sur l'analyse par la théorie de l'élasticité.

Ces modèles permettent de déterminer les sollicitations que subissent les matériaux de chaussée et la plateforme sous l'effet du trafic. Ils constituent une aide précieuse, lors de la conception, dans le choix des dimensions de la superstructure routière.

Il existe aujourd'hui sur le marché un bon nombre de modèles de calcul des contraintes dans le corps de chaussée. Leur évolution est liée à une recherche continue de la représentativité dans la modélisation du comportement mécanique des chaussées.

Si avec le temps on a eu beaucoup de modèles, c'est parce qu'on cherche à reproduire le plus près possible le fonctionnement mécanique de la structure (chaussée).

3.1 Paramètres de calcul.

Les paramètres intervenant dans différents modèles traduisent les effets de l'environnement, du mode de chargement et du trafic.

3.1.1 Environnement.

Même sans l'influence du trafic les structures de chaussée

peuvent se dégrader avec les conditions environnementales :

- érosion pluviale ou éolienne,
- les variations thermiques importantes,
- la remontée de la nappe, etc.

La teneur en eau des sols détermine leurs propriétés, la température a une influence marquée sur les propriétés des liants hydrauliques ou hydrocarbonés. Ces conditions d'environnement varient constamment au cours de la durée de vie d'une chaussée et d'une façon aléatoire.

L'influence de l'eau est prise en compte dans l'adoption du module de déformation pour le sol-support dans les calculs. Mais le problème est en fait assez compliqué, car les interactions entre paramètres sont nombreuses: pluviométrie, évaporation, efficacité du réseau de drainage, nature du sol-support, variations de profondeur de la nappe etc.

Il y a aussi le phénomène de gel, dans le climat tempéré, qui a un grand effet sur la tenue de la chaussée.

Comme on l'a annoncé précédemment, les propriétés des matériaux bitumineux sont largement influencées par la température. Cette sensibilité se traduit par:

- * une réaction élastique aux sollicitations très brèves et à basse température;
- * un fluage aux chargements lents, à température élevée
 - . abaissement des contraintes admissibles,
 - . accroissement de contraintes de traction sous l'effet de la dilatation des couches supérieures,
 - . moins bonne résistance du revêtement au

cisaillement.

Mais les facteurs n'interviennent que dans le choix de la constitution des différentes couches.

Cependant, dans le cas spécifique du climat tropical, on doit reconnaître que généralement:

- le relief est peu accidenté,
- les écarts de température assez modérés

sont autant de facteurs favorables à la construction routière.

3.1.2_ Modes de chargement.

La chaussée est soumise à des charges répétées avec les passages des véhicules usagers.

La roue d'un véhicule immobile exerce sur la chaussée un effort vertical réparti sur toute la surface de contact du pneu. Cette surface de contact donne l'empreinte d'une ellipse quasi rectangulaire qui peut être assimilée à un cercle de rayon a . Donc dépendamment des essieux à roue unique ou à roues jumelées la chaussée est soumise à une charge uniformément répartie, q , sur un cercle ou sur deux cercles de rayon a , distants de d . Généralement $d = 3a$.

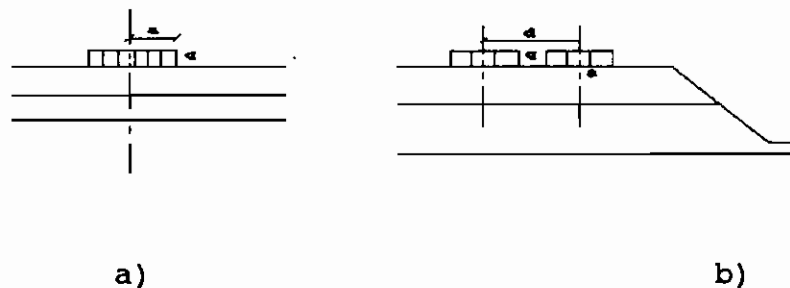


Fig 3.1 Chargement de la chaussée

3.1.3_ Trafic

Le trafic et notamment le trafic de poids lourds, est l'un des paramètres prépondérants du dimensionnement des chaussées; il est donc nécessaire de l'analyser en terme d'agressivité vis-à-vis des chaussées sur lesquelles il circule.

Caractériser le trafic, c'est déterminer le nombre n_{eq} d'essieux standards équivalent à l'ensemble des essieux simples et tandems considérés; le trafic cumulé équivalent du trafic devant réellement circuler sur la chaussée pendant sa durée de vie.

Le nombre n_{eq} est fonction:

- du trafic à la mise en service exprimé en nombre de poids lourds ou en nombre d'essieux;
- de la durée de vie considérée et du taux de croissance.

Selon les méthodes pratiques de dimensionnement, c'est le nombre de cycles n_{eq} ou le trafic à la mise en service qui constitue le paramètre d'entrée.

* Le trafic à la mise en service fait l'objet de mesures ou de prévisions par suite de comptages routiers. Il est souvent exprimé en nombre de poids lourds.

* Durée de service et taux de croissance du trafic:

La durée de service traduit le choix d'une stratégie de dimensionnement, en tenant compte du facteur économie. C'est en principe, la période pendant laquelle on n'aura pas à effectuer d'entretien structurel, les seules opérations nécessaires étant celles liées aux caractéristiques superficielles (adhérence, uni). Le choix de la durée de service s'accompagne d'hypothèses

sur l'état final de la chaussée (à la fin de cette durée).

3.2 Modèles de la mécanique des chaussées. [2,4,5,6]

Le niveau de sollicitation de la structure de chaussée est évalué à l'aide d'un modèle; la suite des équations du modèle représente les interactions entre les divers paramètres gouvernant le fonctionnement de la structure.

Pour rester réalistes les modèles ont évolué avec les structures de chaussées. Cette évolution n'aurait pas été profonde si la généralisation de l'emploi des ordinateurs n'avait permis de résoudre des problèmes physiques d'une grande complexité, à l'aide des méthodes numériques (différences finies, éléments finis) ou en redécouvrant des modèles mathématiques qui n'avaient pas connu un développement par le passé.

Mais il n'existe pas encore à l'heure actuelle de méthode théorique de calcul de chaussées tout à fait satisfaisante. Ceci est dû d'une part à la complexité des phénomènes physiques et d'autre part à la difficulté des solutions mathématiques de ce problème même simplifié. Il est cependant intéressant d'étudier les différentes tentatives qui ont été faites dans ce domaine.

3.2.1_ Modèle de BOUSSINESQ (1885)

Dans le cas des chaussées souples, Boussinesq considère un massif homogène isotrope élastique, semi-infini sollicité par une charge verticale uniformément répartie selon une forme circulaire en surface. Donc ce modèle n'est utilisé que pour le calcul des superstructures monocouches. La contrainte verticale σ_z est

maximale à l'aplomb du cercle de chargement. A la profondeur z elle prend la valeur:

$$\sigma_z = q \left[1 - z^3 / (a^2 + z^2)^{3/2} \right]$$

où q est la charge répartie

a, le rayon du cercle de la charge.

Les autres contraintes le long de l'axe z sont:

$$\sigma_r = \sigma_\theta = q \left[1/2 - A + A^{3/2} + (1-A)\mu \right]$$

$$r_z r = 0$$

$$\text{avec } A = 1 / [1 + (a/z)^2]^{1/2}$$

μ = coefficient de Poisson.

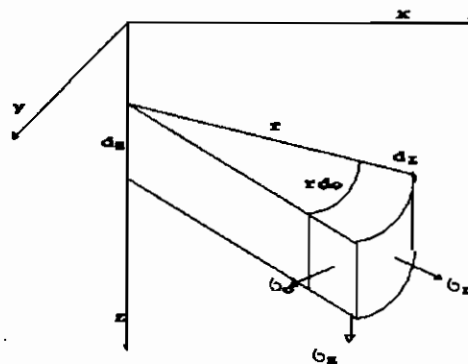


Fig 3.2 Contraintes sur un élément du massif (avec symétrie axiale)

En un point quelconque, la contrainte verticale σ_z est donnée par:

$$\sigma_z = q \left[B - \frac{t}{\pi [t^2 + (1+\rho)^2]} \right]^{1/2} \left[\frac{t^2 - 1 + \rho^2}{t^2 + (1-\rho)^2} E(k) + \frac{1-\rho}{1+\rho} \pi_0(k, n) \right]$$

où $E(k)$ = intégrale elliptique complète de seconde espèce.

$\pi_0(k, n)$ = intégrale elliptique complète de troisième espèce.

$$B = \begin{cases} 1 & \text{si } r < a \\ 1/2 & \text{si } r = a \\ 0 & \text{si } r > a \end{cases}$$

$$\rho = r/a ; \quad t = z/a ; \quad k = \{4\rho/[t^2 + (\rho+1)^2]\}^{1/2} ;$$

$$n = -4\rho/(1+\rho)^2$$

Comme on le voit, la solution purement analytique est déjà relativement compliquée.

3.2.2 Modèle de HOGG (1938)

Le modèle est schématisé sur la figure 3.4.

La chaussée représentée par une plaque mince est posée sur un massif infini de type Boussinesq (E_2, μ_2).

Les déplacements verticaux w de la fibre neutre de la plaque satisfont à l'équation de Lagrange pour les plaques minces:

$$D \cdot \hat{\Delta}^2 w = p \quad (1)$$

avec $D = \frac{E_1 H^3}{12(1-\mu_1^2)}$ caractéristique de la rigidité de la plaque;

H : épaisseur de la plaque;

p : somme des pressions verticales, soit $p = \sigma_z - q$ à l'aplomb du cercle de charge et σ_z à l'extérieur de ce cercle.

L'opérateur $\hat{\Delta}^2$ est le double laplacien en coordonnées polaires:

$$\hat{\Delta}^2 = \left(\frac{d^2}{dr^2} + \frac{d}{rdr} \right)^{(2)}$$

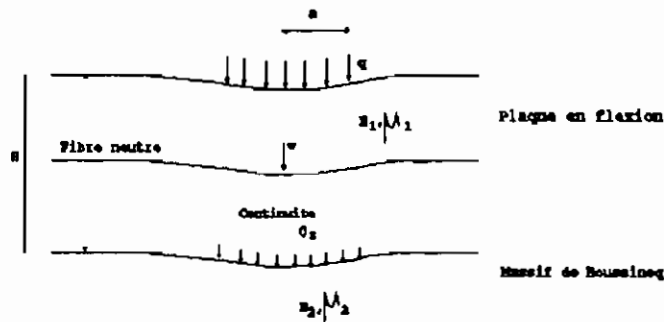


Fig 3.3 Modèle de Hogg

Le problème consiste à déterminer les deux inconnues que sont le déplacement w de la plaque et la contrainte verticale σ_z sur le massif.

En assimilant les déplacements verticaux de la fibre neutre de la plaque aux déplacements verticaux du massif; ceux-ci étant reliés à la contrainte sur le massif par les formules de Boussinesq. Les moments fléchissants de la plaque sont déduits des déplacements verticaux par les formules usuelles des plaques.

En supposant que la chaussée glisse parfaitement sur son support on trouve comme contrainte à la base (contrainte de traction) de la plaque (pour $\mu_1 = \mu_2 = \mu$),

$$\sigma_t = \frac{1+\mu}{2} \frac{q H E_1}{a E_2} \int_0^{\infty} \frac{x J_1(x)}{1 + \frac{H^3 E_1 x^3}{6a^3 E_2}} dx$$

où $J_1(x)$ est la fonction de Bessel de première espèce.

3.2.3 Le modèle de WESTERGAARD (1926)

La complexité relative du modèle de Hogg provient de la nature même du massif de Boussinesq qui supporte la chaussée.

Westergaard a adopté pour le sol une autre hypothèse pour faciliter les calculs. Le modèle considère la plaque mince comme la chaussée et assimile le sol-support à un assemblage de ressorts dont le déplacement en un point (w) est proportionnel à la pression verticale v en ce point; soit $v = k.w$

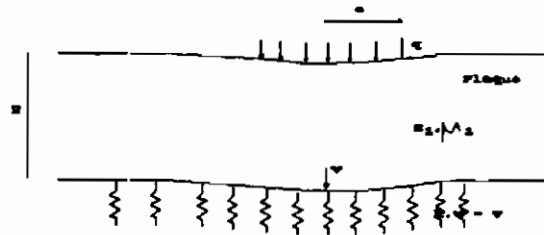


Fig 3.4 Modèle Westergaard

- v : contrainte verticale sur le massif;
- w : déplacement vertical de la plaque;
- k : module de réaction du sol-support.

L'équation de Lagrange pour les plaques minces s'écrit:.

$$\hat{\Delta}^2(w) = p(r)/D$$

où p désigne la somme des efforts de flexion à la distance r du centre de la plaque:

$$p(r) = \begin{cases} v(r) - q & \text{si } r < a \\ v(r) & \text{si } r > a \end{cases}$$

On définit la fonction de la charge à la surface par:

$$q(r) = \begin{cases} q & \text{si } r < a \\ 0 & \text{si } r > a \end{cases}$$

$$\Rightarrow \hat{\Delta}^2(w(r)) = \frac{v(r)-q(r)}{D} \quad \left(D = \frac{E_1 H^3}{12(1-\mu_1^2)} \right)$$

$$v(r) = k.w(r) \implies \Delta^2(w(r)) + k.w(r)/D = q(r)/D$$

Donc on a une équation différentielle du quatrième ordre en $w(r)$.

3.2.4 Le modèle de BURMISTER (1943)

Burmister a franchi une étape importante en donnant la solution du calcul des contraintes et des déformations, dans un massif multicouche élastique [2] semi-infini sous une charge circulaire. Jeuffroy-Bachelez ont élaboré des tableaux et des abaques pour des modèles tricouches avec une simplification des hypothèses de Burmister.

Burmister a traité le problème général à n couches schématisé comme suit:

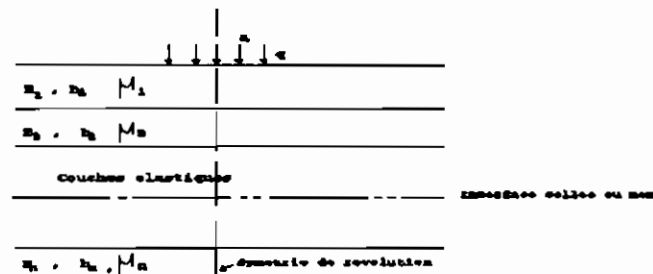


Fig 3.5 Modèle de Burmister

Dans le modèle de Burmister toutes les couches sont traitées comme des solides homogènes élastiques et isotropes. Les interfaces entre couches peuvent comporter des couches collées ou des couches décollées. Le cas des charges multiples (roues jumelées, essieux tandem ou tridem, remorques ou mille-pattes) peut être traité par superposition des effets des charges élémentaires.

La limitation principale de ce modèle est le fait, comme dans le modèle de Boussinesq ou de Hogg, que les couches sont infinies en plan. Le modèle ne permet donc pas d'aborder les effets de bord (charge au bord d'une fissure, ou au bord de la chaussée).

Néanmoins, les bords sont en général suffisamment éloignés de la bande de roulement, pour que l'hypothèse de couches infinies puisse être le plus souvent adoptée; ce qui n'est pas le cas pour les dalles de béton pour lesquelles il est nécessaire de faire des calculs pour des charges en bord ou en angle de dalle.

Par ailleurs, c'est cette hypothèse de couches infinies qui facilite grandement les calculs.

L'état de contraintes et de déformations d'un élément d'un milieu continu en coordonnées cylindriques est déterminé par:

$\sigma_r, \sigma_\theta, \sigma_z$: contraintes normales.
$\tau_{\theta z}, \tau_{rz}, \tau_{r\theta}$: contraintes de cisaillement
$\epsilon_r, \epsilon_\theta, \epsilon_z$: déformations linéaires
$\delta_{\theta z}, \delta_{rz}, \delta_{r\theta}$: déformations angulaires

Si u, v, w sont respectivement des déplacements radial, tangential et la direction z on a:

$$\begin{aligned} \epsilon_r &= du/dr & \epsilon_\theta &= u/r + (1/r)dv/d\theta & \epsilon_z &= dw/dz \\ \delta_{r\theta} &= (1/r)dv/d\theta + dv/dr - v/r & \delta_{rz} &= du/dz + dw/dr \\ \delta_{z\theta} &= dv/dz + (1/r)dw/d\theta \end{aligned}$$

Les conditions d'équilibre s'il n'y a pas de forces massiques s'écrivent:

$$\frac{d\sigma_r}{dr} + \frac{1}{r} \frac{d\tau_{r\theta}}{d\theta} + \frac{d\tau_{rz}}{dz} - \frac{\sigma_r - \sigma_\theta}{r} = 0$$

$$d\tau_{rz}/dr + (1/r)d\tau_{\theta z}/d\theta + d\sigma_z/dz + \tau_{rz}/r$$

$$d\tau_{r\theta}/dr + (1/r)d\sigma_{\theta}/d\theta + d\tau_{\theta z}/dz + 2\tau_{r\theta}/r$$

où d est l'opérateur de dérivée partielle.

Avec la considération de la symétrie axiale (donc contraintes indépendantes de l'angle θ voir Fig 3.6 b)), on obtient:

$$\frac{d\sigma_r}{dr} + \frac{d\tau_{rz}}{dz} - \frac{\sigma_r - \sigma_{\theta}}{r} = 0$$

$$d\tau_{rz}/dr + d\sigma_z/dz + \tau_{rz}/r$$

$$\epsilon_r = du/dr \quad \epsilon_{\theta} = u/r \quad \epsilon_z = dw/dz$$

$$\delta_{rz} = du/dz + dw/dr$$

La condition de compatibilité avec la loi de Hooke donne:

$$\Delta\sigma_r - 2/r^2 (\sigma_r - \sigma_{\theta}) + 1/(1+\mu)d^2 (\sigma_r + \sigma_{\theta} + \sigma_z)/dr^2 = 0$$

$$\Delta\sigma_{\theta} - 2/r^2 (\sigma_r - \sigma_{\theta}) + 1/(1+\mu)\frac{1}{r} d(\sigma_r + \sigma_{\theta} + \sigma_z)/dr = 0$$

Et il est démontré que la fonction de contraintes ϕ satisfait les conditions d'équilibre et de compatibilité si et seulement si:

$$(d^2/dr^2 + 1/r d/dr + d^2/dz^2)(d^2\phi/dr^2 + 1/r d\phi/dr + d^2\phi/dz^2) = \\ = \Delta\Delta(\phi) = 0$$

Soit $\Delta^2\phi(r,z) = 0$ où Δ^2 est le double laplacien.

Donc la résolution d'un problème d'élasticité en coordonnées cylindriques avec symétrie axiale, se réduit à la recherche de fonctions de tension à double laplacien nul:

$$\Delta^2\phi(r,z) = 0 \quad 2.4.1$$

On peut exprimer alors les contraintes et déplacements en termes de fonction de contraintes ϕ comme:

Contraintes

$$\sigma_z = d[(2-\mu)\Delta\phi - d^2\phi/dz^2]/dz$$

$$\sigma_r = d(\mu\Delta\phi - d^2\phi/dr^2)/dz$$

$$\sigma_{\theta} = d(\mu \Delta \phi - 1/r(d\phi/dr))/dz$$

$$\tau_{rz} = d[(1-\mu)\Delta\phi - d^2\phi/dz^2]/dr \quad 2.4.2$$

Déplacements

$$w = (1+\mu)/E \cdot [(1-2\mu)\Delta\phi + d^2\phi/dr^2 + 1/r(d\phi/dr)]$$

$$u = -(1+\mu)/E (d^2\phi/dr dz)$$

Pour les systèmes multicouches chacune des équations 2.4.2 doit être résolue et par les conditions aux limites (interface supérieure et interface inférieure de la couche).

Considérant seulement le cas de symétrie axiale, on a trouvé:

$$\phi^i = J_0(mr) [(A_i + B_i z) e^{mz} + (C_i + D_i z) e^{-mz}] \quad 2.4.3$$

où i se réfère à la couche i du système de couches;

$J_0(mr)$ est la fonction de Bessel de première espèce d'ordre 0;

A_i, B_i, C_i et D_i sont des constantes d'intégration qui peuvent être déterminées à partir des conditions aux limites.

Pour le cas particulier du massif de Boussinesq $\phi = B(r^2+z^2)^{1/2}$

En substituant l'équation 2.4.3 dans les équations 2.4.2 on a les contraintes et déplacements à l'interface i qui sont:

Contraintes

$$\sigma_z^i = m^2 J_0(mr) \{ (1-2\mu_i) (B_i e^{mz} + D_i e^{-mz}) - m[(A_i + B_i z) e^{mz} + (C_i + D_i z) e^{-mz}] \}$$

$$\sigma_{\theta}^i = 2\mu_i m^2 J_0(mr) (B_i e^{mz} + D_i e^{-mz}) + m^2 \frac{J_1(mr)}{mr} \{ B_i e^{mz} + D_i e^{-mz} + m[(A_i + B_i z) e^{mz} - (C_i + D_i z) e^{-mz}] \}$$

$$\sigma_r^i = m^2 J_0(mr) \{ (1 + 2\mu_i) (B_i e^{mz} + D_i e^{-mz}) + m[(A_i + B_i z) e^{mz} + (C_i + D_i z) e^{-mz}] \} - m^2 \frac{J_1(mr)}{mr} \{ B_i e^{mz} + D_i e^{-mz} + m[(A_i + B_i z) e^{mz} + (C_i + D_i z) e^{-mz}] \}$$

$$\tau_{rz}^i = m^2 J_1(mr) (2\mu_i (B_i e^{mz} - D_i e^{-mz}) + m[(A_i + B_i z) e^{mz} + (C_i + D_i z) e^{-mz}])$$

Déformations

$$u^i = \frac{1 + \mu_i}{E_i} m J_1(mr) (B_i e^{mz} + D_i e^{-mz} + m[(A_i + B_i z) e^{mz} + (C_i + D_i z) e^{-mz}])$$

$$w^i = \frac{1 + \mu_i}{E_i} m J_0(mr) (2(1 - \mu_i) (B_i e^{mz} - D_i e^{-mz}) - m[(A_i + B_i z) e^{mz} + (C_i + D_i z) e^{-mz}])$$

(qui constituent les équations 2.4.4)

où J_n est la fonction de Bessel de première espèce d'ordre n

Les $4n$ conditions aux limites à vérifier, pour déterminer les constantes d'intégration, se décomposent comme suit:

- A la surface

$$\sigma_z = q(r) = \begin{cases} q & \text{si } r \leq a \\ 0 & \text{si } r > a \end{cases}$$

$$\tau_{rz} = 0$$

(2 conditions)

- A l'infini

$A_n = 0$ et $B_n = 0$; sinon les paramètres dont les formules sont données précédemment donneraient infini comme valeurs. Ce qui est irréel. Donc on a également 2 conditions.

- Aux (n-1) interfaces

Interface	Collee	Decollee
Conditions	$\sigma_z^{i-1} = \sigma_z^i$	$\sigma_z^{i-1} = \sigma_z^i$
	$\tau_{rz}^{i-1} = \tau_{rz}^i$	$\tau_{rz}^{i-1} = 0$
	$u^{i-1} = u^i$	$\tau_{rz}^i = 0$
	$v^{i-1} = v^i$	$v^{i-1} = v^i$
	4 conditions	4 conditions

Ce qui donne $4(n-1)$ conditions.

Soit au total $2 + 2 + 4(n-1) = 4n$ conditions.

n: nombre de couches.

L'interface i est celle qui limite la i^{eme} couche en bas.

Du fait que les équations 2.4.3 sont des fonctions linéaires en ϕ , la fonction de contraintes:

$$\phi(r, z) = \int_0^{\infty} \phi(r, z, m) dm$$

constituera aussi la solution de ces équations lorsque le système est sujet à la charge: $p(m)J_0(mr)dm$

La transformation inverse de Hankel d'ordre zéro donne:

$$\frac{1}{2} \left[f(r-0) + f(r+0) \right] = \int_0^{\infty} m [rf(r)J_0(mr)dr] J_0(mr) dm$$

Ainsi le paramètre de charge est:

$$p(m) = m \int_0^{\infty} rf(r)J_0(mr)dr$$

Dans le cas particulier de $r=a$ on a $f(r) = 1/2$

Et $p(m) = aJ_1(ma)$

Ainsi les contraintes et déplacements à n'importe quelle

profondeur dans la couche i sont obtenus par:

$$\begin{aligned} \sigma_z^i &= a \int_0^\infty J_1(ma) \sigma_z^i dm \\ \sigma_r^i &= a \int_0^\infty J_1(ma) \sigma_r^i dm \\ \sigma_\theta^i &= a \int_0^\infty J_1(ma) \sigma_\theta^i dm \\ \tau_{rz}^i &= a \int_0^\infty J_1(ma) \tau_{rz}^i dm \\ u^i &= a \int_0^\infty J_1(ma) u^i dm \\ w^i &= a \int_0^\infty J_1(ma) w^i dm \end{aligned} \quad (2.4.5)$$

où σ_z^i , σ_r^i , σ_θ^i , τ_{rz}^i , u^i et w^i sont les contraintes et déplacements à l'interface i , obtenus des équations 2.4.4.

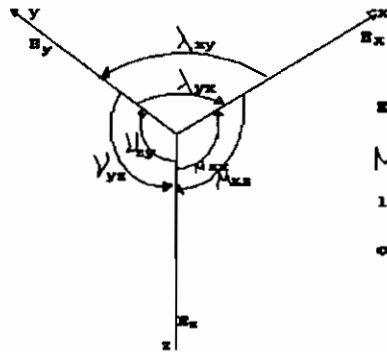
3.2.5_ Modèle tenant compte de l'anisotropie

Ceci a été l'oeuvre de Frans VAN CAUWELAERT dans sa thèse de doctorat Es sciences techniques à l'Ecole Polytechnique Fédérale de Lausanne (E.P.F.L).

Ce modèle est exactement celui de Burmister, mais tenant compte de la théorie d'anisotropie.

L'anisotropie d'un matériau est caractérisée par la variation de ses propriétés d'une direction à l'autre. Elle résulte de son état de compacité.

-Un massif anisotrope en trois dimensions est représenté par:



E : module d'élasticité

μ_{xy} : coefficient de Poisson caractérisant la déformation linéaire selon y pour une contrainte normale selon x

-Le matériau orthotrope qui est matériau anisotrope avec plan de symétrie est caractérisé par:

$$E_x = E/m ; E_y = E/n ; E_z = E$$

$$\mu_{xy} = L/m ; \mu_{xz} = \mu/m ; \mu_{yz} = Nu/n$$

$$\mu_{yx} = L/n ; \mu_{zx} = \mu ; \mu_{zy} = Nu$$

où n et m sont des degrés d'anisotropie.

Quant au matériau anisotrope à isotropie transverse, c'est-à-dire qui est isotrope dans un plan (généralement le plan horizontal) et anisotrope dans la direction perpendiculaire à ce plan il est caractérisé par:

$$E_x = E/m = E_y ; E_z = E;$$

$$\mu_{xy} = L ; \mu_{xz} = \mu/m = \mu_{yz};$$

$$\mu_{yx} = L ; \mu_{zx} = \mu ; \mu_{zy} = Nu .$$

Mais Van CAUWELAERT montre qu'il n'y a pas d'erreur appréciable en prenant $Nu = \mu$. Et que la plupart des problèmes (surtout en mécanique des chaussées peut être résolue en considérant que les matériaux sont anisotropes à isotropie transverse; c'est-à-dire isotropes dans le plan horizontal et anisotropes dans la direction perpendiculaire à ce plan.

Et on peut prendre sans grande $L = \mu = Nu$.

N.B : Faute d'existence de certains caractères grecs comme λ et ν on a ces derniers respectivement par L et Nu

Le degré d'anisotropie, qui rend compte de l'effet de concentration des contraintes verticales dans les massifs chargés en surface, est défini comme le rapport entre le module élastique perpendiculaire au plan isotrope au module élastique dans ce plan est traduit par: $m = E/(E/m)$.

Les relations entre déformations et contraintes sont données par:

$$\epsilon_x = (\sigma_x - \mu_{xy}\sigma_y - \mu_{xz}\sigma_z)/E_x$$

$$\epsilon_y = (\sigma_y - \mu_{yz}\sigma_z - \mu_{yx}\sigma_x)/E_y$$

$$\epsilon_z = (\sigma_z - \mu_{zx}\sigma_x - \mu_{zy}\sigma_y)/E_z$$

$$\delta_{xz} = \tau_{xz}/G_{xz}$$

Les relations entre déformations et déplacements sont les mêmes qu'à la section 2.4).

Et V. CAUWELAERT montre que la fonction de contraintes est donnée par:

$$\Delta\Delta_1\phi = (d^2/dr^2 + 1/r d/dr + d^2/dz^2)[d^2\phi/dr^2 + 1/r d\phi/dr + (1/s^2)d^2\phi/dz^2] = 0$$

où $s = 1$ 'indice d'anisotropie = $[(m-\mu^2)/(m^2-\mu^2)]^{1/2}$

$$\text{Ainsi } \phi_i = J_0(tr) [A_i e^{tz} - B_i e^{-tz} + C_i e^{stz} - D_i e^{-stz}]$$

En considérant que ϕ ne contient pas des fonctions trigonométriques

$$\phi = J_0(tr) * f(z).$$

Et en utilisant les relations 2.4.2 on obtient des équations semblables à celles de 2.4.4)

En considérant les mêmes conditions aux limites que dans le cas 2.4),

on a: $A_n = 0$ et $C_n = 0$ à l'infini.

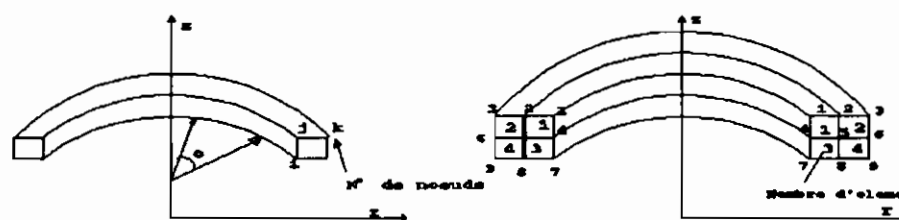
3.2.6 Modèle d'éléments finis.

En 1949 Newmark stipulait que l'utilisation du modèle d'éléments finis offre certains avantages: pas d'ambiguïté concernant les conditions aux limites, les variations des dimensions et des propriétés physiques peuvent être facilement traitées etc... Et beaucoup de chercheurs se sont tournés vers ce modèle.

Mais bien que la nouvelle technologie permet l'utilisation d'ordinateurs de grande capacité, les méthodes d'éléments finis ne peuvent être brillamment traitées.

L'application de ce modèle aux systèmes multicouches a été effectuée par Wilson et Barksdale.

Le modèle d'éléments finis développé par Barksdale utilise un système de coordonnées cylindriques représenté par (r, θ, z) . La structure de chaussée est considérée comme un assemblage d'éléments circulaire ayant une section rectangulaire dans le plan rz .



a)

b)

Fig 3.6 Modèle d'éléments finis.

Les noeuds i, j, k et l d'un élément sont numérotés dans le sens

horaire.

Le vecteur déplacement global du noeud i est donné par:

$\{\delta_i\} = \{v_i, w_i\}$ où v_i est le déplacement dans la direction r et w_i celui dans la direction z

Le vecteur déplacement d'un élément est donné alors par:

$$\{\delta_i\} = \{\delta_i, \delta_j, \delta_k, \delta_l\} \quad (2.6.1)$$

Ainsi le vecteur déplacement pour les n éléments du système peut être présenté par:

$$\{\delta\} = \{\delta_1, \delta_2, \dots, \delta_n\} \quad (2.6.2)$$

De la même manière le vecteur de forces externes nodales est représenté par:

$\{F_i\} = \{F_r, F_z\}$ pour le noeud i.

$\{F\} = \{F_1, F_2, \dots, F_n\}$ pour le système.

L'analyse d'équilibre pour chaque système donne la relation force-déplacement:

$\{F\} = [K]\{\delta\}$ où [K] est la matrice de flexibilité.

Pour un matériau isotrope, la relation entre contraintes et déformations est donnée, avec la théorie d'élasticité par:

$$\begin{bmatrix} \sigma_z \\ \sigma_r \\ \sigma_o \\ \tau_{rz} \end{bmatrix} = \frac{E(1-\mu)}{(1+\mu)(1-2\mu)} \begin{bmatrix} 1 & \frac{\mu}{1-\mu} & \frac{\mu}{1-\mu} & 0 \\ 1 & 1 & \frac{\mu}{1-\mu} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{1-2\mu}{2(1-\mu)} \end{bmatrix} \begin{bmatrix} \epsilon_z \\ \epsilon_r \\ \epsilon_o \\ \delta_{rz} \end{bmatrix}$$

qui peut être simplifiée sous la forme:

$$\{\sigma\} = [D]\{\epsilon\}$$

avec $\{\sigma\}$: vecteur de contraintes

$\{\epsilon\}$: vecteur de déplacements

$[D]$: matrice élastique.

La déformation à n'importe quel point de l'élément peut être déterminée par la relation:

$$\{\epsilon_z, \epsilon_r, \epsilon_\theta, \delta_{rz}\} = \{dw/dz, dv/dr, dw/dz + dv/dr\}$$

(δ_{rz} déformation angulaire)

qui, après substitution et simplification donne $\{\epsilon\} = [B]\{\delta_n\}$

avec $\{\delta_n\}$: vecteur déplacement pour le noeud n;

$[B]$: matrice de relation entre déformation totale et déplacement.

La matrice de flexibilité pour chaque élément est déterminée selon Wilson, par:

$$[K] = \int_{\text{vol}} [B]^T [D] [B] dv$$

L'intégration porte sur tout le volume de l'élément.

Les forces concentrées sont considérées dans le vecteur $\{F\}$.

Les forces réparties sont converties en forces concentrées équivalentes appliquées aux noeuds.

On considère une charge uniformément répartie q appliquée normalement à la surface $r-\theta$ de l'élément n (fig 3.7 a)). La force concentrée équivalente F_z^k dans la direction z sur le noeud k est égal au travail de la charge q quand $w_k=1$ avec tous les autres déplacements des noeuds j et k , nuls.

$$F_j^k = 2\pi q \int_{r_j}^{r_k} \frac{(r-r_j)}{(r_k-r_j)} r dr$$

$$\Rightarrow F_z^k = \frac{2\pi q (r_j^3/6 - r_k \cdot r_j^2/2 + r_k^3/3)}{(r_k - r_j)}$$

on a: $\{F\} = [K]\{\delta\}$.

3.3 Confrontation des modèles

En plus des insuffisances individuelles déjà signalées notons que les modèles précédents supposent les matériaux de chaussée élastiques et isotropes, sauf pour le dernier où les matériaux sont à anisotropie transverse. Ils supposent également que des tractions excessives ne se produisent pas dans des couches dénuées de cohésion. Ces hypothèses sont assez bien vérifiées dans le cas des chaussées en béton, mais beaucoup moins dans le cas des chaussées souples.

La distorsion principale vient du comportement viscoélastique des sols argileux et des enrobés bitumineux. La réponse de ces milieux aux sollicitations brèves provoquées par le passage des véhicules est très différente de la réponse sous une charge statique.

En ce qui concerne le modèle d'éléments finis, la solution des problèmes d'élasticité est possible par suite des hypothèses simplificatrices sur la façon dont les déplacements et contraintes varient dans chaque élément du modèle. Les hypothèses sont d'autant plus justifiées que les dimensions de l'élément sont réduites; c'est-à-dire que la discrétisation est poussée.

Et plus la discrétisation est poussée plus la taille des matrices à manipuler est grande. Il y a donc nécessité d'ordinateurs de grande capacité.

Par ailleurs ce modèle, comme les autres, suppose la charge statique et une symétrie axiale de cette charge appliquée sur le plan horizontal. Il ne tient pas compte des effets de bords et de coins.

<p>DIMENSIONNEMENT DES SUPERSTRUCTURES ROUTIERES.</p>
--

Le dimensionnement des chaussées intègre l'étape essentielle qui est celle de la conception de la structure. A cette étape on définit le meilleur agencement possible des divers matériaux dont on dispose en tenant compte de leur résistance aux actions du trafic et de l'environnement. C'est à cette étape qu'on détermine:

- l'épaisseur de chaque couche à mettre en place,
- si une couche doit être traitée ou non,
- si on admet l'apparition des fissures de retrait à la surface de la chaussée,
- s'il faut réduire le nombre de couches ou pas.

4.1 Concept de sécurité et critères de rupture.

Le concept de sécurité en route dépend en fait de l'objectif que l'on fixe à la chaussée que l'on veut calculer et ne peut être déterminé qu'après l'établissement des critères de rupture.

La définition de l'objectif fixé à la chaussée est nécessaire pour déterminer les contraintes ou déformations admissibles dans les matériaux.

Dire que l'on veut calculer une chaussée pour une durée de vie donnée, dix ans par exemple, n'est pas très opérationnelle; car

on ne laisse jamais une chaussée se dégrader jusqu'à la fin de sa durée de vie, sans travaux d'entretien qui prolongent cette durée de vie. Mais si l'on s'en tient au seul objectif d'obtenir une durée de vie fixée, on définit la fin de la vie d'une chaussée qui n'est manifestement pas une ruine généralisée de la structure.

On peut conventionnellement fixer cette fin de la vie de la chaussée comme le moment où la proportion de la surface de la chaussée présentant des dégradations (fissurations, ornières...) dépasse une valeur fixée à l'avance. On peut aussi se donner comme objectif que la probabilité d'apparition des dégradations avant une période de temps donnée, considérée comme durée de vie recherchée, soit inférieure à une valeur fixée.

A partir de l'objectif on considère que la rupture intervient quand la contrainte admissible dans le matériau est atteinte ou si la déformation engendrée par les sollicitations dépasse la déformation admissible. Et ce n'est que la marge prise par rapport à l'état admissible (ou critique) qui constitue la sécurité.

Certains auteurs utilisent un coefficient de sécurité dit " à la rupture " qui est le rapport de la contrainte de rupture à la contrainte effective: $F.S = \sigma_r / \sigma_{eff}$. La contrainte effective étant celle appliquée au matériau.

La connaissance des propriétés mécaniques des matériaux et du mode de sollicitations appliquées à la chaussée permet de fixer les contraintes maximales à ne pas dépasser pour éviter la rupture.

La mécanique de la rupture montre que les contraintes limites dans un matériau dépendent de la géométrie de la pièce et du mode d'application des sollicitations.

Alors qu'en construction routière on ne connaît avec précision, ni

les propriétés des matériaux utilisés, ni le mode d'application des sollicitations, ce n'est qu'avec des approximations qu'on définit les différents paramètres.

Par ailleurs, dans le dimensionnement d'une chaussée on doit rester très près de la rupture. Car, contrairement aux autres ouvrages de Génie Civil en général, le calcul n'a pas pour objectif de rendre l'apparition des dégradations fort improbable, du fait que la dégradation de la chaussée n'a jamais le caractère catastrophique que peut avoir la rupture d'un pont par exemple. Et on cherche à réduire dans la mesure du possible le coût de construction et d'entretien.

La vérification mécanique porte sur les quatre (4) formes de rupture que sont:

- * la fissuration de l'enrobé;
- * la fatigue en flexion des couches traitées aux liants hydrauliques;
- * le poinçonnement du sol d'assise;
- * l'apparition en surface des déformations excessives, produisant des ornières ou des défauts inacceptables.

4.2 Calcul des épaisseurs.

Le but principal recherché dans la construction routière est d'avoir une structure plus économique et plus harmonieuse du point de vue de son évolution dans le temps. Pour ce faire les épaisseurs de différentes couches doivent être déterminées, de telle manière que les formes de rupture énoncées précédemment n'apparaissent pas avant le nombre de cycles de chargement admissible.

4.3 Lois de fatigue [2]

La loi de fatigue de chaque matériau permet de déterminer le nombre de cycles de chargement que peut supporter ce matériau sans subir une forme spécifique de rupture. C'est dans cette loi qu'interviennent le trafic et l'environnement.

4.3.1 Rupture en fatigue par flexion répétée des matériaux traités aux liants hydrauliques.

Les essais de fatigue, pour les matériaux traités aux liants hydrauliques sont effectués en flexion sur des éprouvettes prismatiques trapézoïdales en répétant pour chacune d'elles l'application d'un niveau de contrainte préalablement choisi. Pour ces matériaux on peut mettre la courbe de fatigue sous la forme: $\sigma = \sigma_0(1-a \log N)$

où $\sigma_0 = R_F$ (résistance à la traction déterminée par l'essai en flexion) résistance à la rupture au premier chargement,

σ = contrainte à la rupture après N cycles de chargement,

a = la pente de la courbe;

N = le nombre de cycles à la rupture.

De multiples travaux ont montré qu'on peut admettre $a=1/20$ pour la plupart de ces matériaux.

$$\text{Donc } Nf = 10^{20(1-\sigma/\sigma_0)}$$

4.3.2 Matériaux bitumineux

4.3.2.1. Rupture par fissuration de l'enrobé

Ici on peut adopter sans erreur appréciable,

$$\log \epsilon_m = \log(1.6 \cdot 10^{-3})^{-0.21} \log N$$

où N = nombre de cycles de chargement

ϵ_N = déformation admissible,
comme loi de fatigue.

Le nombre de cycles avant fissuration est alors:

$$N_e = (1.6 \cdot 10^{-3} / \epsilon_N)^{1/0.21}$$

avec $\epsilon_N = [\sigma_r^1 - \mu(\sigma_z^1 + \sigma_\theta^1)] / E_1$

$\epsilon_z, \sigma_r, \sigma_\theta$ définis dans le chapitre 3

En supposant σ_r très peu différent de σ_θ $\epsilon_N = [\sigma_r - \mu(\sigma_z + \sigma_r)]$

4.3.2.2 Orniérage limite (= 1.5 cm)

La courbe de fatigue peut être représentée par:

$$\epsilon = \frac{115(\sigma_z - \sigma_r)}{E} (t/10^3)^{0.25}$$

où $\sigma_z - \sigma_r$ est le déviateur des contraintes verticale et radiale au point de coordonnées (z,r)

ϵ : déformation en compression répétée du revêtement hydrocarboné,

E: module de l'enrobé,

t: temps de mise en charge (secondes).

Le calcul de l'orniérage, qui est en fait le déplacement vertical d'un point de la surface, nécessite l'étude de la variation du déviateur de contraintes en fonction de la profondeur. En supposant le module et le temps de mise en charge constants, la déformation varie de la même façon que le déviateur, lorsque ce dernier est positif.

Aux points où le déviateur est négatif on considère que les déformations sont nulles. Ainsi le calcul de l'orniérage prend en compte l'épaisseur h sur laquelle le déviateur est positif.

Fig 4.1 Variation du déviateur en fonction de la température.

Dans le cas de la température minimale extrême θ_1 , où on peut avoir une répartition triangulaire (voir fig 4.1 a)), on calcule la déformation à la base du revêtement soit:

$$\epsilon_0 = \frac{115(\sigma_r^1 - \sigma_r^1)}{E_1} (t/10^3)^{0.25}$$

avec E_1 module du revêtement à la température θ_1 .

L'orniérage est alors:

$$d = (h/2)\epsilon_0 \text{ à la température } \theta_1$$

Par ailleurs dans le cas de température élevée extrême θ_2 , où on peut avoir une répartition trapézoïdale, le calcul de l'orniérage porte sur toute l'épaisseur de la couche bitumineuse (fig 4.1 b)).

On calcule une déformation moyenne:

$$\epsilon_m = \frac{115[(\sigma_1 - \sigma_2) + (\sigma_r^1 - \sigma_r^1)]}{E_1} (t/10^3)^{0.25}$$

E_1 : module à la température θ_2

L'orniérage est alors:

$$d = H_1 \epsilon_m$$

H_1 : épaisseur de la couche bitumineuse.

- Calcul du temps de mise en charge.

En prenant V_m comme vitesse moyenne du trafic considéré le temps de passage d'un essieu est donné par:

$$t_p = D/V_m$$

D: diamètre de l'empreinte de charge ($D = 2 \cdot 12.5 = 25$ cm)

Le temps de mise en charge total avant rupture est alors:

$$t = N t_p = N D/V_m . \text{ Soit } t = N D/V_m$$

En prenant par exemple $V_m = 70$ km/h = 19.44 m/sec

$$\text{on a: } t = N 0.25/19.44 = 0.0129 \text{ sec}$$

- Répartition du trafic selon les saisons

On subdivise la saison en une période froide, où on observe la faible température, θ_1 (plus fort module), et une période chaude où on a la température élevée θ_2 (plus faible module E).

En admettant que dans l'année on a observé une fraction α_1 du trafic pendant la saison froide et l'autre fraction ($\alpha_2=1-\alpha_1$) pendant la période chaude l'orniérage total serait:

$$d_{\text{tot}} = d_{(\theta_1)} + d_{(\theta_2)} = \frac{h}{2} \frac{115 (\sigma_z^1 - \sigma_r^1)}{E_1} \left(\frac{0.25 \alpha_1 N}{V_m 10^3} \right)^{0.25} +$$

$$\frac{h}{2} \frac{115 [(\sigma_1 + \sigma_2) + (\sigma_z^1 - \sigma_r^1)]}{E_1} \left(\frac{0.25 \alpha_2 N}{V_m 10^3} \right)^{0.25}$$

V_m en m/sec.

$$\text{Soit } d_{\text{tot}} = 3.446 N^{0.25} \left[\frac{h (\sigma_z^1 - \sigma_r^1)}{E_{1(\theta_1)}} (\alpha_1/V_m)^{0.25} + \right.$$

$$\left. \frac{h_1 ((\sigma_1 + \sigma_2) + (\sigma_z^1 - \sigma_r^1)) (\alpha_2/V_m)^{0.25}}{E_{1(\theta_2)}} \right]$$

$$\text{d'où } N_{\text{orn}} = 7.091513 \cdot 10^{-3} \frac{d_{\text{tot}}^{(2)}}{d_{\text{tot}}^{(1)}} \left[\frac{h (\sigma_z^1 - \sigma_r^1)}{E_{1(\theta_1)}} (\alpha_1/V_m)^{0.25} + \right.$$

$$\left. \frac{h_1 ((\sigma_1 + \sigma_2) + (\sigma_z^1 - \sigma_r^1)) (\alpha_2/V_m)^{0.25}}{E_{1(\theta_2)}} \right]^{-4}$$

4.3.3_ Matériaux non traités

Pour le sol-support, indépendamment des essais de laboratoire, divers auteurs ont établi des lois de fatigue qui sont certainement inexactes mais suffisantes à l'usage. On peut citer les formules:

$$\epsilon_z = 21600 \cdot 10^{-6} N^{-0.28} \quad \text{Université de Nottingham}$$

$$\epsilon_z = 28000 \cdot 10^{-6} N^{-0.25} \quad \text{Formule Shell (1977)}$$

$$\epsilon_z = 11000 \cdot 10^{-6} N^{-0.23} \quad \text{Formule CRR}$$

Mais généralement on retient $\epsilon_z = 21000 \cdot 10^{-6} N^{-0.24}$

Certains auteurs font intervenir la contrainte σ_z sur le sol et en particulier:

$$\text{Heukelom:} \quad \sigma_z = 0.006 E_p / (1 + 0.7 \log N)$$

$$\text{Formule danoise:} \quad \sigma_z = 0.152 E \cdot N^{-0.307}$$

Le CEBTP propose pour le calcul du nombre de cycles admissibles avant poinçonnement de la plate-forme la formule semi-empirique de Kerkhoven et Dormon, jugée très sécuritaire pour les latérites:

$$\sigma_{zadm} = 0.3 \text{CBR} / (1 + 0.7 \log N) \quad (\text{Bars})$$

$$N_{sol} = 10^{[0.3 (\text{CBR}/\sigma_z) - 1] / 0.7}$$

avec N: nombre de cycles, E_p : module dynamique.

Des corrélations entre le module de rigidité des sols avec certains paramètres usuels tels que le CBR ont été proposées: c'est ainsi qu'on trouve dans certains ouvrages:

$$E_{\text{dynamique}} = 100 \text{CBR} \quad (\text{Bars}) \quad (\text{Shell}) \quad [2]$$

$$E_{\text{statique}} = 50 \text{CBR} \quad (\text{Bars}) \quad \text{pour les matériaux grossiers (insensibles à l'eau)} \quad [2]$$

$$E_{\text{statique}} = 30 \text{CBR} \quad (\text{Bars}) \quad \text{pour les matériaux avec fines} \quad [1]$$

$$E = 65 \text{CBR}^{0.65} \quad (\text{Bars}) \quad \text{Jeuffroy Bachelez.}$$

4.4) Détermination de la durée de vie de la superstructure routière

Comme dans le paragraphe précédent, on calcule d'abord, pour les autres formes de rupture, le nombre de cycles admissible à la température de θ_1 et à θ_2 . On pondère ensuite ces deux résultats avec la loi de Miner [2].

Si N^{θ_1} est le nombre de cycles admissible à θ_1 ,

N^{θ_2} , celui à θ_2 et

N , le nombre de cycles global (durée de vie).

Miner montre que

$$\frac{\alpha_1 N}{N_{\theta_1}} + \frac{\alpha_2 N}{N_{\theta_2}} \leq 1$$

Et on a à l'égalité

$$N = \frac{N_{\theta_1} N_{\theta_2}}{\alpha_1 N_{\theta_2} + \alpha_2 N_{\theta_1}}$$

Donc l'application de cette loi aux formes de rupture par poinçonnement de la plate-forme, par fissuration de l'enrobé ou par flexion répétée des couches liées permet l'obtention (en nombre de cycles de chargement) des durées de vie relatives à ces modes de rupture.

On choisit le minimum de ces nombres comme durée de vie déterminante qu'on confronte au trafic (nombre de cycles) supposé au départ.

$$N_{\text{det}} = \min (N_{\text{orn}}, N_e, N_f, N_{\text{sol}})$$

- Prévision du trafic

Les moyens de comptage et de pesée permettent de réduire le trafic composite, à la répétition d'un nombre n_{eq} d'essieux standard (P_0); les lois d'équivalence entre charges font intervenir la nature des différentes configurations des véhicules (essieux isolés, tandem et tridem) et aussi la nature des structures de chaussées. Ces lois sont du type $(P/P_0)^\alpha$

Dans le cadre de l'essai AASHO, Liddle propose $c = \frac{(\frac{P}{8.2})^\alpha}{(8.2)}$

avec $\alpha=4$ pour les structures souples, $4 < \alpha < 8$ pour les chaussées rigides.

c'est le coefficient d'équivalence.

La loi de Liddle est valable pour n'importe quel essieu standard.

Par exemple, un transport exceptionnel de 480t (16 essieux de 30t) sur une structure souple a sur cette chaussée le même effet que $\frac{16(30/8.2)^4}{2(9.5/8.2)^4} = 796$ camions de 19t (2 essieux de 9.5t)

Le nombre de cycles durant la durée de vie est déterminé par:

$$N = 365t_1 \frac{((1+i)^n - 1)}{i} \quad \text{pour une croissance exponentielle.}$$

$$N = 365nt_1 \frac{(2 + (n-1)i)}{2} \quad \text{pour une croissance linéaire.}$$

avec t_1 : trafic moyen journalier de mise en service qui peut être converti en n_{eq}

n : nombre d'années (durée de vie)

i : taux de croissance annuel du trafic

**PROGRAMME, ZOVROUTE, DE CALCUL DES SUPERSTRUCTURES
ROUTIERES AVEC LE MODELE DE BURSMISTER,
UTILISANT L'ELASTICITE ISOTROPE .**

5.1) Raisons du choix du modèle [2,4,5]

Le choix de ce modèle pour la programmation s'explique par les raisons telles que :

La complexité du modèle de Hogg qui provient de la nature même du massif de Boussinesq qui supporte la chaussée. Le comportement du sol est infiniment plus complexe.

Pour le modèle de Westergaard, les transformations inverses de Hankel qui permettent de remonter aux valeurs des fonctions recherchées (déplacements, contraintes) n'est pas facile. La difficulté s'exprime par la complexité du calcul de l'intégrale pour laquelle on ne connaît pas toujours de résultat sous forme d'une fonction simple. Donc bien que ce modèle prenne en compte les bords, son adaptation à l'informatique reste difficile .

Alors que pour le modèle de Burmister, pour toute valeur m , on peut trouver explicitement les fonctions du système de couches. Avec les valeurs des contraintes et déplacements $(\sigma(m), w(m))$ obtenus on conduit numériquement l'intégration 2.4.5 pour obtenir les contraintes et déplacements réels. Et comme nous l'avons signalé

précédemment les bords sont en général suffisamment éloignés de la bande de roulement pour que l'hypothèse de couches (horizontales) infinies soit souvent adoptée; sauf pour le cas particulier de dalles de béton pour lesquelles il est nécessaire d'effectuer des calculs pour des charges en bord ou en angle de dalle.

La non adoption du modèle anisotrope de Burmister réside dans la difficulté, et le manque de matériels nécessaires, du moins dans nos pays en voie de développement, pour obtenir les degrés d'anisotropie .

5.2 Différentes étapes du programme [3]

Le programme, qui constitue le logiciel ZOVROUTE, qu'on trouvera en annexes est relatif à des superstructures routières à n couches $n \leq 5$, et couvre donc la majorité des cas pratiques. Il permet de formuler soit l'hypothèse de la continuité parfaite des couches soit le glissement entre la première et la deuxième couche et/ou entre le glissement entre la deuxième et la troisième couche.

Le système d'équations linéaires est obtenu à partir des $4n$ conditions aux limites.

Une interface est parfaitement glissante si la différence entre les caractéristiques des couches adjacentes est importante.

Après l'introduction des caractéristiques des différentes couches et le mode de chargement, le programme démarre selon des étapes bien déterminées. (Voir organigramme en annexes.)

5.2.1 L'établissement du système d'équations linéaires.

A partir des équations 2.4.4, on prend en compte les

conditions aux limites pour établir le système.

On pouvait trouver les fonctions en utilisant les formes matricielles pour chaque couche. Mais cette méthode entraîne la recherche des inverses de matrices, ce qui nécessite beaucoup d'opérations; donc d'erreurs de troncature relativement importantes.

A la surface du système de couches, $z=0$, la contrainte verticale (normale) doit être égale à la pression exercée par la pneumatique sur un rayon a et à zéro (0) à l'extérieur de ce rayon. La contrainte de cisaillement y est nulle.

Avec la transformation de Hankel cette contrainte à la surface est définie par:

$$\sigma_0 = -a \cdot \int_0^{\infty} J_0(mr) J_1(ma) dm.$$

à q près. Donc on a:

$$\begin{aligned} a \cdot \int_0^{\infty} J_1(ma) \{ -m^2 J_0(mr) [-mA_1 - (2\mu_1 - 1)B_1 + mC_1 - (2\mu_1 - 1)D_1] \} dm = \\ = -a \cdot \int_0^{\infty} J_0(mr) J_1(ma) dm. \end{aligned}$$

Ce qui est équivalent à:

$$m^2 [-mA_1 - (2\mu_1 - 1)B_1 + mC_1 - (2\mu_1 - 1)D_1] = -1$$

$$\tau_0 = 0 \implies mA_1 + 2\mu_1 B_1 + mC_1 - 2\mu_1 D_1 = 0$$

Conditions d'interfaces.

$$\sigma_z^i = \sigma_z^{i+1} \implies \sigma_z^i - \sigma_z^{i+1} = 0 \iff$$

$$[-me^{mz_i} A_i + (1 - 2\mu_i - mz_i) e^{mz_i} B_i + me^{-mz_i} C_i + (1 - 2\mu_i + mz_i) e^{-mz_i} D_i] -$$

$$[-me^{mz_{i+1}} A_{i+1} + (1 - 2\mu_{i+1} - mz_{i+1}) e^{mz_{i+1}} B_{i+1} + me^{-mz_{i+1}} C_{i+1} + (1 - 2\mu_{i+1} + mz_{i+1}) e^{-mz_{i+1}} D_{i+1}] = 0$$

$$w_i = w_{i+1} \iff$$

$$(1 + \mu_i) / E_i \cdot [-me^{mz_i} A_i + (2 - 4\mu_i - mz_i) e^{mz_i} B_i - me^{-mz_i} C_i + (2 - 4\mu_i + mz_i)$$

$$e^{-mz_i} D_i] - (1 + \mu_{i+1}) / E_{i+1} \cdot [me^{mz_{i+1}} A_{i+1} + (2 - 4\mu_{i+1} - mz_{i+1}) e^{mz_{i+1}} B_{i+1} +$$

$$-me^{-mz_i}C_{i+1} + (2-4\mu_{i+1}+mz_i)e^{-mz_i}D_{i+1}] = 0$$

qui est transformée en:

$$(1+\mu_i) [-me^{mz_i}A_i + (2-4\mu_i-mz_i)e^{mz_i}B_i - me^{-mz_i}C_i + (2-4\mu_i+mz_i)e^{-mz_i}D_i] - (1+\mu_{i+1})(E_i/E_{i+1}) [me^{mz_i}A_{i+1} + (2-4\mu_{i+1}-mz_i)e^{mz_i}B_{i+1} - me^{-mz_i}C_{i+1} + (2-4\mu_{i+1}+mz_i)e^{-mz_i}D_{i+1}] = 0$$

- Interface Collée

$$\tau_{rz}^i = \tau_{rz}^{i+1} \iff$$

$$[me^{mz_i}A_i + (2\mu_i+mz_i)e^{mz_i}B_i + me^{-mz_i}C_i + (mz_i-2\mu_i)e^{-mz_i}D_i] - [me^{mz_i}A_{i+1} + (2\mu_{i+1}+mz_i)e^{mz_i}B_{i+1} + me^{-mz_i}C_{i+1} + (mz_i-2\mu_{i+1})e^{-mz_i}D_{i+1}] = 0$$

$$u^i = u^{i+1} \iff$$

$$(1+\mu_i)/E_i \cdot [me^{mz_i}A_i + (mz_i+1)e^{mz_i}B_i - me^{-mz_i}C_i + (1-mz_i)e^{-mz_i}D_i] - (1+\mu_{i+1})/E_{i+1} \cdot [me^{mz_i}A_{i+1} + (mz_i+1)e^{mz_i}B_{i+1} - me^{-mz_i}C_{i+1} + (1-mz_i)e^{-mz_i}D_{i+1}] = 0$$

qui est transformée en:

$$(1+\mu_i) [me^{mz_i}A_i + (mz_i+1)e^{mz_i}B_i - me^{-mz_i}C_i + (1-mz_i)e^{-mz_i}D_i] - (1+\mu_{i+1})(E_i/E_{i+1}) \cdot [me^{mz_i}A_{i+1} + (mz_i+1)e^{mz_i}B_{i+1} - me^{-mz_i}C_{i+1} + (1-mz_i)e^{-mz_i}D_{i+1}] = 0$$

- Interface décollée.

$$\tau_{rz}^i = 0 \iff$$

$$me^{mz_i}A_i + (2\mu_i+mz_i)e^{mz_i}B_i + me^{-mz_i}C_i + (mz_i-2\mu_i)e^{-mz_i}D_i = 0$$

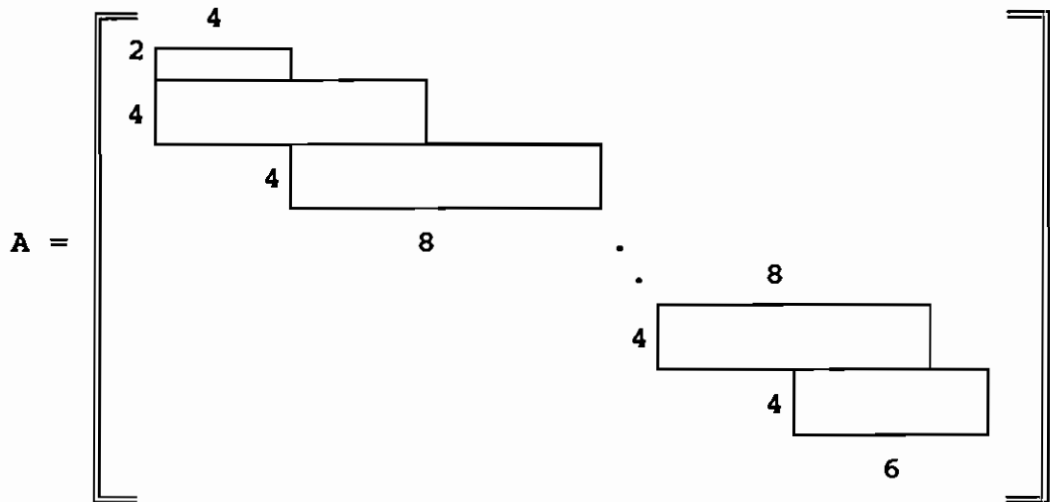
$$\tau_{rz}^{i+1} = 0 \iff$$

$$me^{mz_i}A_{i+1} + (2\mu_{i+1}+mz_i)e^{mz_i}B_{i+1} + me^{-mz_i}C_{i+1} + (mz_i-2\mu_{i+1})e^{-mz_i}D_{i+1} = 0$$

où les A_i , B_i , C_i , D_i sont des inconnues.

Si on a n comme nombre de couches le nombre d'inconnues sera $4n-2$ parce que ayant 4 inconnues par couche et $A_n = C_n = 0$.

La matrice du système obtenu est de la forme:



Notons que les équations précédentes sont obtenues après simplification par m ou m^2 . Mais il est tout à fait intéressant de remarquer que toutes les fonctions sont nulles si m est nul.

La résolution du système obtenu est effectuée avec la méthode d'élimination de Gauss-Jordan. Cette méthode est bien appropriée compte tenu du fait que la matrice du système est diagonale par bloc, donc relativement creuse. D'ailleurs on ne peut pas utiliser la relaxation du fait que la matrice n'est pas symétrique.

Suite à la détermination des inconnues les contraintes ou déplacements, fonctions de m sont calculées et intégrées.

5.2.2_ Calculs d'intégrales

Il existe plusieurs méthodes pour calculer l'intégrale d'une fonction connue. Entre autres: méthodes des trapèzes, de Simpson, de Gauss, de Romberg...

Le principal inconvénient des méthodes des trapèzes et de Simpson réside dans le fait que l'on ne connaît pas la précision du résultat. A nombre d'intervalles n égal, la méthode de Gauss est

plus précise que ces méthodes. Malheureusement avec n intervalles, la précision est inacceptable pour les fonctions autres que les polynômes. L'usage de la méthode de Gauss est donc réservée à ces derniers. L'approximation est exacte pour les polynômes de degré inférieur à $2n$; les résultats sont alors obtenus très rapidement et avec une grande précision.

La méthode de Romberg permet de choisir à priori la précision; les itérations se font alors jusqu'à l'obtention de la précision demandée. Elle repose sur une utilisation originale de la méthode des trapèzes et est utilisée pour n'importe quelle fonction.

L'intégration de toute fonction se fait par intervalles si cette dernière admet des zéros dans l'intervalle déterminé par les bornes d'intégration. Les intégrations (2.4.5) doivent donc être effectuées par intervalles; d'autant plus que les fonctions qui y interviennent sont des fonctions biharmoniques.

Les fonctions à intégrer sont du type: $J_i(mr)J_i(ma)F(m, z)$

($i=0$ ou 1). Pour un z et un r donnés $F(m, z)$ garde un signe constant. Donc les bornes des différentes intégrales sont définies par les racines des fonctions de Bessel de première espèce d'ordre 0 et d'ordre 1 .

Donc une brève étude des fonctions de Bessel de première espèce s'avère nécessaire.

Ces fonctions sont définies par:

$$J_n(x) = \sum_{r=0}^{\infty} \frac{(-1)^r (x/2)^{2r+n}}{r! \Gamma(n+r+1)}$$

où Γ est la fonction Gama. Si $n+r$ est entier $\Gamma(n+r+1)=(n+r)!$

Ainsi

$$J_0(x) = \sum_{r=0}^{\infty} \frac{(-1)^r (x/2)^{2r}}{r! \Gamma(r+1)} = \sum_{r=0}^{\infty} \frac{(-1)^r (x/2)^{2r}}{r! r!}$$

$$J_1(x) = \sum_{r=0}^{\infty} \frac{(-1)^r (x/2)^{2r+1}}{r! \Gamma(1+r+1)} = \frac{1}{2} \sum_{r=0}^{\infty} (-1)^r (x/2)^{2r+1} / (r!(r+1)!)$$

Il est à remarquer que $J_1(x)/x$ tend vers $\frac{1}{2}$ quand x tend vers 0.

L'approximation asymptotique de ces fonctions entre 1 et l'infini est donnée par:

$$J_n(x) = (2/\pi x)^{1/2} [P_n(1/x) \cos(X_n) - 1/x Q_n(1/x) \sin(X_n)]$$

où l est un argument non nul arbitraire choisi.

P_n et Q_n sont des polynômes d'ordre inférieur ou égal à l .

$$X_n = x - (2n+1)\pi/4.$$

Cette forme asymptotique facilite énormément l'adaptation à l'informatique, des fonctions de Bessel.

Ces fonctions sont infiniment définies et dérivables. Et il est démontré que:

$$|J_0(x)|^2 + 2|J_1(x)|^2 + \dots + 2|J_n(x)|^2 + \dots = 1$$

Donc quelque soit x , $|J_0(x)| \leq 1$ et $|J_n(x)| \leq 1/\sqrt{2}$ ($n \geq 1$)

Les racines et les courbes de J_0 et J_1 dans l'intervalle $[0, 30]$ sont fournies en annexes.

5.3 Caractéristiques du logiciel " ZOVROUTE"

Les contraintes constituent les principaux paramètres déterminés dans ce logiciel établi en langage Turbo Pascal 5.5.

Le module du dimensionnement se base sur ces contraintes pour déterminer le nombre de cycles de chargement admissible pour une structure routière donnée.

Les entrées du logiciel sont définies par:

- le nombre de couches;
- l'épaisseur de chaque couche si l'objectif est le calcul des fonctions pour une structure déjà en place. Sinon,

dans le cas de dimensionnement, il y a des épaisseurs de départ pour lesquelles le programme détermine, sur la base des lois de fatigue, le nombre de cycles. Et dépendamment des ressources disponibles, l'opérateur peut modifier, à sa convenance, les épaisseurs de différentes couches jusqu'à avoir la durée de vie (ou nombre de cycles à la rupture) voulue. La corrélation, entre le module et le CBR, utilisée pour l'application des lois de fatigue, pour les matériaux non traités, est celle proposée par Shell: $E = E_p = 100\text{CBR}$. Le CBR intervenant dans cette relation est celui obtenu après quatre jours d'imbibition des éprouvettes. Donc la formule $N_{\text{sol}} = 10^{[0.3 (\text{CBR}/\sigma_z) - 1]/0.7}$ devient

$$N_{\text{sol}} = 10^{[0.003 (E/\sigma_z) - 1]/0.7}$$

- conditions d'interfaces;
- le module d'élasticité de chaque couche;

A ce niveau, on doit connaître, dans le cas du dimensionnement, le module de la couche bitumineuse en période froide comme en période chaude. Ainsi, avec les pourcentages du trafic pendant ces périodes on peut, comme montré précédemment, déterminer le nombre de cycles satisfaisant pour cette couche. Et, compte tenu du fait que les contraintes transmises à une couche sont fonction des matériaux des couches sous-jacentes, le nombre de cycles ainsi déterminé influera sur ceux des autres couches.

- le coefficient de Poisson de chaque couche.
- les coordonnées du point (excentricité r , profondeur z)

où on voudrait déterminer les fonctions.

Lorsque le logiciel est chargé en mémoire, le menu principal apparaît à l'écran. Ce menu permet de choisir le type d'opération qu'on désire effectuer.

- Types d'opérations.

Il y en a de quatre sortes:

- * Calcul de contraintes
- * Dimensionnement
- * Rappel de fichier des résultats d'une structure donnée.
- * Retour au système Dos

Lorsque l'une des deux premières est choisie, l'introduction des données s'effectue comme suit:

- Nombre de couches.

Désigné par n dans le rapport mais par nc dans le programme. Le nombre maximum que peut accepter le logiciel est 5.

- Modules d'élasticité (Bars).

Désignés par E_i dans le rapport mais par e_i dans le programme. E_i apparaît autant de fois que le nombre de couches.

Dans le cas de calcul de contraintes, les modules sont entrés pour une température donnée. Alors que pour le dimensionnement l'entrée de deux modules: le module de la période froide et celui de la période chaude de l'année, est nécessaire pour la couche bitumineuse (revêtement).

- Glissement interface?

Introduire 0 s'il n'y a pas de glissement, 1 s'il en a un: 1/2 ou 2/3, 2 s'il en a aux interfaces 1/2 et 2/3.

- si on introduit 1 au stade précédent il apparaît

Interface 1/2 o/n ?

Introduire 0 qui signifie que le glissement est à l'interface 1/2 (interface entre les couches 1 et 2); sinon c'est à l'interface 2/3.

- **Essieu standard. ?**

Introduire la charge à l'essieu (en tonnes).

Apparaît alors **Roues jumelées o/n?**. Introduire 0 si l'essieu est à roues jumelées, n sinon.

Le rayon standard d'empreinte de pneu étant de 12.5 cm la pression de la roue sur la surface de la structure est calculée par la formule

$q = 100 \cdot 10 \cdot (\text{charge}/4)/A$ si l'essieu est à roues jumelées, sinon $q = 100 \cdot 10 \cdot \text{charge}/(2A)$.

$A = \pi \cdot 12.5^2 \cdot q$ en bars (1 bar = 10² kPa 1 kg = 10 N).

- **Coefficients de Poisson.**

Désignés par μ_i dans le rapport, ils le sont par mmi dans le programme. $\mu_i = ?$ apparaît autant de fois que le nombre de couches.

En plus de ces données, s'ajoute pour:

* **Calcul des contraintes**

+ **Épaisseurs de couches.**

S'il y a n couches on introduit n-1 épaisseurs. La dernière étant la plateforme considérée comme ayant une épaisseur infinie (suffisamment grande).

+ Coordonnées du point à caractériser.

Ceci concerne le cas de calcul de contraintes.

- Nombre de points à caractériser n_p

Ce nombre est limité à cinq (5) compte-tenu du fait qu'un nombre élevé nécessiterait beaucoup de temps.

-Profondeurs (cm) des points?

-Excentricité (cm)?

Elle est la distance par rapport à l'axe de la roue considérée.

*** Dimensionnement**

+ Module de la première couche, considérée bitumineuse, en période chaude (période 2).

Ceci suppose que le dimensionnement du logiciel n'est que pour les structures souples.

+ Pourcentages du trafic en période froide, α_1 , et en période chaude α_2 .

+ Si la couche de base est traitée au ciment la loi de fatigue à utiliser est celle par flexion répétée. Donc il y a nécessité d'entrer la résistance à la rupture, σ_r ou R_f (déterminée au laboratoire) à l'essai en flexion .

+ Vitesse moyenne du trafic (km/h).

L'option **Rappel de fichier** permet de rappeler directement un fichier existant. Après le rappel de ce fichier, le logiciel Zovroute affiche le contenu à l'écran. Mais en aucun cas, on ne peut agir sur les paramètres affichés.

La dernière opération permet le retour au système

d'exploitation Dos.

Le stockage comme l'impression sont possibles après l'entrée des données.

Le démarrage du programme commence par la lecture des racines nécessaires dépendamment des coordonnées r , z ou h_1 . h_1 étant l'épaisseur de la première couche.

Puisque les produits $m*r$ et $m*a$ constituent les arguments des fonctions de Bessel intervenant dans les opérations, les racines à lire vont alors de zéro à $m*r$ ou $m*a$. Ainsi si r est inférieur à $a = 12.5$ cm, les racines sont lues jusqu'à $10*a = 125$. Les racines des fonctions de Bessel tabulés sont compris entre 0 et 600. Puisque, si l'excentricité entrée supérieure à 12.5, le calcul des contraintes aux points d'excentricité allant jusqu'à 200 cm.

Le dimensionnement prend en compte les cas critiques. Il s'agit, pour le cas de roue simple, des points situés sur l'axe de la roue et aux interfaces.

Au bout d'un certain temps le programme sort avec l'affichage des caractéristiques du système de couches et des résultats.

Dans le cas du dimensionnement, l'opérateur a la possibilité d'agir sur les épaisseurs des couches jusqu'à avoir la situation optimale (du point de vue économique que structural).

Ensuite il y a la possibilité d'imprimer ou de stocker ces résultats. Le signe négatif indique que la contrainte est une compression. Dans le cas contraire elle est en traction.

5.3.1_ Contraintes sous roues jumelées

Les composantes du tenseur de contraintes sont obtenues par superposition (somme algébrique des contraintes). Ceci est d'autant plus justifié que nous travaillons avec l'hypothèse que les matériaux sont élastiques et isotropes.

Pour le calcul des contraintes, du fait que l'entre_axe des roues est de 37.5 cm ($d = 3a$), la superposition se fait avec les contraintes à r et à $|37.5-r|$ cm. r étant l'excentricité donnée. Donc les opérations, dans le cas de roues jumelées prendront pratiquement, le double du temps que dans le cas de roue simple.

Concernant le module de dimensionnement, il ne serait pas bien établi si on ne détermine pas les points critiques.

Il est évident que les interfaces constituent les positions critiques mais dans le cas de roues jumelées nous allons voir, à travers des exemples sur le cas: roue simple, lequel des deux cas: deux fois la contrainte à l'axe du jumelage ($r = 18.75\text{cm}$) et superposition de contraintes à 0 et à 37.5 cm donnerait le cas critique.

5.3.2_ Interprétation des résultats.

Tout d'abord, il faut noter que les contraintes de surface sont obtenues avec une moindre précision que les autres. Le calcul de ces contraintes prend, pratiquement, les trois quart (3/4) du temps d'exécution. Elles sont obtenues à 10^{-2} près malgré le nombre d'itérations qu'on y a attribué (voir 5.5).

On pouvait chercher à obtenir le même degré de précision que dans les autres cas. Mais l'exécution fera des heures voire des jours.

A priori on sait qu'en dehors du cercle de chargement, la contrainte verticale est nulle.

Les contraintes sont celles à la base des couches ou à une profondeur quelconque. Elles sont, en général, pour la charge: $q = 6.5 \text{ kg}$, essieu à roue simple (6.62 bars). Et pour une structure à 4 couches avec adhérence entre ces dernières.

La couche 0 constitue la surface; couche 1, base du revêtement.

L'analyse des résultats (en annexes) montrent que si le module de la couche de revêtement est élevé par rapport à celui de la couche de base, il est fort probable que la contrainte radiale ou tangentielle à la base de cette couche (1) soit positive (traction). Donc il y a risque de fissuration de l'enrobé.

Cette situation s'explique par le fait que d'une part l'axe neutre (en flexion), de la section transversale, du système est un peu plus balancé vers le haut que par rapport au cas où toutes les couches auraient le même module et d'autre part la couche de revêtement se comporte comme une dalle par rapport aux autres: donc transforme une grande partie de charges transmises en contraintes de flexion. Ce qui est, approximativement, le caractère d'une structure rigide.

On observe le même cas de figure, cette fois pour la couche de base, dans le cas où c'est le module de cette dernière qui est plus élevé. Dans ce cas la fissuration, s'il doit y en avoir, sera profonde et se propagera, dans le temps, vers la surface.

1	1
2	2
3	3
Mêmes modules	Module 1 élevé

Position de l'axe neutre de flexion

Les résultats montrent que les contraintes diminuent d'une face à l'autre d'une couche. Les contraintes verticales diminuent avec la profondeur.

Les contraintes radiale et tangentielle, sur une couche, sont plus élevées si l'épaisseur de la couche est faible ou si son module est élevé.

En général les contraintes diminuent avec le rayon.

Si une interface est glissante, la contrainte, radiale ou tangentielle, supérieure est en tension alors que celle inférieure est en compression.

Comme on devait s'y attendre les contraintes radiale et tangentielle sont identiques à l'axe de la roue simple. En effet, comme le montre la figure 3.2 si $r = 0$ le massif se réduit à l'axe z . Et dans ce cas la direction de la contrainte tangentielle est confondue à la direction radiale.

Par ailleurs l'observation des résultats nous montre que la superposition des contraintes aux interfaces sous l'axe d'une roue ($r = 0$ et $r = 37.5$ cm) donne les cas critiques pour l'essieu à roues jumelées.

Donc dans le cas de dimensionnement les calculs sont alors

effectués à $r = 0$ et à $r = 37.5$ cm pour la période 1 que pour la période 2. Ce qui nécessite quatre (4) fois plus du temps que le calcul des contraintes aux interfaces sous une roue simple. Et l'analyse des résultats dans ce cas montre qu'il suffit d'une modification de l'une des hauteurs pour que le mode de rupture change. En effet pour les deux exemples donnés, dans le premier on voit que le mode de rupture, s'il doit y en avoir, serait la fissuration de la couche de base traitée; alors que dans le second, qui porte une diminution de 1 cm sur le revêtement et une diminution de 5 cm pour la couche de fondation, la forme de rupture qu'on peut avoir est le poinçonnement du sol d'assise; et puis il n'y a pas d'orniérage.

L'analyse des contraintes verticales (bars):

$$E_1=50000 \quad E_2=40000 \quad E_3=5000 \quad E_4=500 \text{ bars}$$

$$\mu_1=0.4 \quad \mu_2=0.4 \quad \mu_3=0.3 \quad \mu_4=0.3$$

$$h_1=5 \quad h_2=15 \quad h_3=30 \text{ cm}$$

Profondeurs (cm)	r=0	r=18.75	r=37.5
0	-6.5945	-0.1027	-0.0207
-5	-5.9396	-0.2204	-0.0188
-20	-0.9500	-0.4542	-0.1326
-50	-0.0728	-0.0988	-0.0765

$$E_1=40000 \quad E_2=50000 \quad E_3=5000 \quad E_4=500 \text{ bars}$$

$$\mu_1=0.4 \quad \mu_2=0.4 \quad \mu_3=0.3 \quad \mu_4=0.3$$

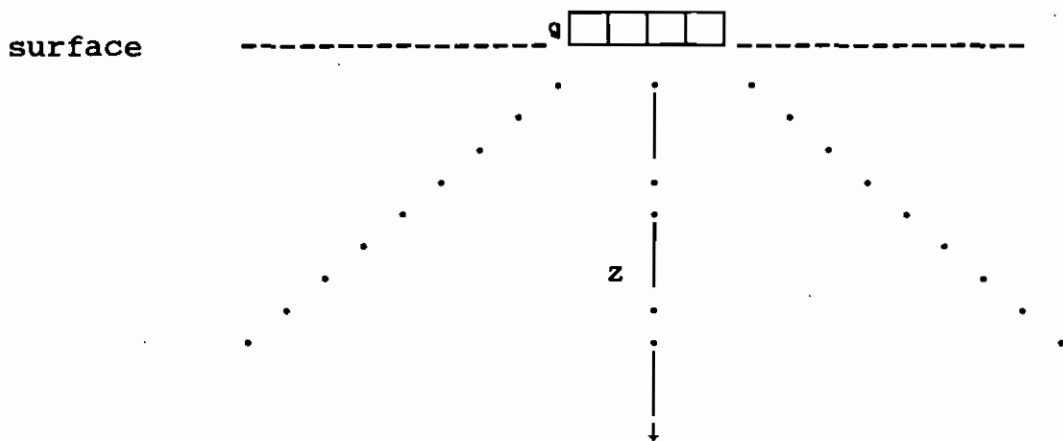
$$h_1=5 \quad h_2=15 \quad h_3=30 \text{ cm}$$

Profondeurs (cm)	r=0	r=18.75	r=37.5
0	-6.5945	-0.1027	-0.0207
-5	-6.0128	-0.1690	-0.0179
-20	-0.8902	-0.4412	-0.1381
-30	-0.0715	-0.0988	-0.0769

nous fait, abstraction faite des contraintes de la surface

($0 < z < 5$ cm), représenter la bulbe de pression, sans trop

d'erreurs comme suit:



5.4 Définition des touches

- **Les touches fléchées** \downarrow \uparrow \leftarrow \rightarrow sont surtout utilisées pour le mouvement du curseur lors des entrées ou pour le défilement des résultats.

Pour le dimensionnement, lors de la modification des épaisseurs, les flèches \leftarrow \rightarrow sont utilisées pour sélectionner l'épaisseur à modifier; celles \uparrow \downarrow sont respectivement utilisées pour augmenter ou diminuer d'une unité, par frappe, l'épaisseur choisie.

- **La touche entrée** est pour valider les choix. Mais il est à noter que, pour le cas du dimensionnement, cette touche ne sera utilisée qu'après avoir porté de modifications sur l'ensemble, des épaisseurs, désiré.

- **Les touches Esc, F2, F4, F7, F8, F10**

Esc: permet, lors des entrées, de revenir à l'écran précédent.

F2: permet, après une opération, de sauvegarder les données ou les résultats dans un fichier.

F4: est utilisée pour imprimer les résultats.

Une autre manière d'imprimer les résultats est d'utiliser

la touche **ImpÉcr** ou **PrtScr** pour imprimer ce qui est à l'écran.

F7: permet de lancer les opérations de calcul de contraintes ou de dimensionnement.

F8: permet de sortir du cadre des entrées de caractéristiques.

F10: permet de revenir au menu principal.

5.5 Limites du programme

Le logiciel Zovroute ne paraît pas imposer des restrictions. Toutefois sa convergence dépend des coordonnées ou de l'épaisseur de la première couche.

En effet si z ou $h_1 > 5$ cm la convergence est relativement rapide. Et le résultat est obtenu avec une bonne précision déjà à partir de $m = 2$ ($12.5 * m = 25$). Mais cette convergence est très lente vers la surface. C'est pourquoi dans ce cas (où z ou $h_1 \leq 5$ cm), $m = 10$ ($12.5 * m = 125$) est adopté. Donc l'intégration sera effectuée de 0 à 125.

Plus r , z , h_1 croissent plus la convergence est rapide.

La principale restriction de ce programme est le nombre de couches qui est limité à 5. Mais c'en n'est vraiment pas une, d'autant plus que dans la pratique il est rare d'avoir une structure routière ayant plus de 4 couches.

Par ailleurs il y a des limitations relatives à la capacité de l'ordinateur.

En effet si d'une part la mémoire de l'ordinateur n'a pas une certaine taille l'exécution du programme peut être entravée. Mais une mémoire vive de **640 Koctets**, qui est d'ailleurs la capacité minimale des ordinateurs courants, suffit largement; d'autre part

si l'ordinateur ne dispose pas de coprocesseur numérique 8087/80287 l'exécution du programme est extrêmement lente.

Cette dernière limitation vient du fait que les fonctions de la matrice du système fluctuent considérablement d'une colonne à une autre. Leurs valeurs peuvent dépasser 10^{+38} ou être inférieures à 10^{-39} qui constituent les limites des variables de type réel en simple précision. Cette fluctuation s'explique par le fait que les produits mz ou mh apparaissent en exponentiel. Déjà quand h_i est de l'ordre de 30 cm la limite de la simple précision est atteinte.

Le choix de la double précision permet non seulement la possibilité d'avoir des couches de grande épaisseur, mais aussi de réduire les erreurs de troncature. Mais malgré ce choix une certaine limite demeure. Cette limite concerne la profondeur de la dernière interface (interface entre la plateforme et la superstructure).

En effet, pour des raisons évoquées dans le paragraphe 5.3.2, dans le calcul des contraintes à une profondeur $z < 5$ cm avec $r < 12.5$ cm ou dans le cas de recherche de nombres de cycles de chargement, cette interface doit se trouver à une profondeur inférieure ou égale à 70 cm. Dans le calcul des contraintes à $z < 5$ cm et $r > 12.5$ cm, elle est à 230 cm au plus. Si $5 < z < 15$ cm, l'interface est à $z = 350$ cm au maximum. Et enfin pour le calcul des contraintes à $z > 15$ cm, cette interface doit se trouver à $z \leq 700$ cm.

CONCLUSION ET RECOMMANDATIONS

Le modèle de Burmister nous a permis de construire un logiciel de calcul de contraintes dans les systèmes multicouches. Il permet aussi, avec introduction des lois de fatigue selon la nature des couches ainsi que les risques acceptés aux différents niveaux, la détermination de la durée de vie d'une chaussée. Mais ce dernier cas ne s'applique qu'aux structures souples.

L'utilisation de ce modèle suppose que les matériaux sont élastiques, homogènes et isotropes et que la charge appliquée est circulaire.

Le logiciel, qui est spécifique au cas de route, peut être facilement adapté à d'autres cas d'analyse de contraintes, en prenant le rayon du cercle de charge comme une variable. Ainsi il pourrait être utilisé, par exemple dans le calcul des contraintes dans une fondation sous poteaux circulaires.

Dans ce programme c'est le rayon normalisé (12.5 cm) qui est considéré comme rayon du cercle de charge.

Outil d'usage tout à fait facile et assurant des résultats de précision suffisante, ZOVROUTE est utilisable sur les micro-ordinateurs qu'on trouve maintenant non plus dans les services sophistiqués, mais dans tous les bureaux de secrétariat.

Il ne saurait se substituer au catalogue des structures qui, outre les calculs, intègre les résultats des observations sur les chaussées existantes.

L'analyse des résultats, obtenus sous l'hypothèse d'adhérence parfaite, permet de constater que plus le module de la couche est élevé par rapport au module de la couche de base, il y a risque de traction à la base de la couche. Or, dans la construction routière on cherche à éviter souvent cette traction pour ne pas avoir une fissuration structurale.

Ceci nous amène à dire que, dans la conception d'une structure souple, on doit choisir le bitume dont le module ne s'écarte pas trop, avec la période, du module de la couche de base.

Ainsi en choisissant un matériau ayant la moyenne des modules des deux périodes extrêmes de la couche bitumineuse, on réduirait un tant soit peu le risque de fissuration qu'on peut avoir; surtout en période froide.

Mais pour rester réaliste, Zovroute doit faire l'objet d'un calage qui consiste à l'observation d'une structure souple qu'il aurait permis de dimensionner.

Et nous ne doutons pas que certains prendront le relais et adapter le cas de calcul des contraintes dans les structures rigides à l'informatique.

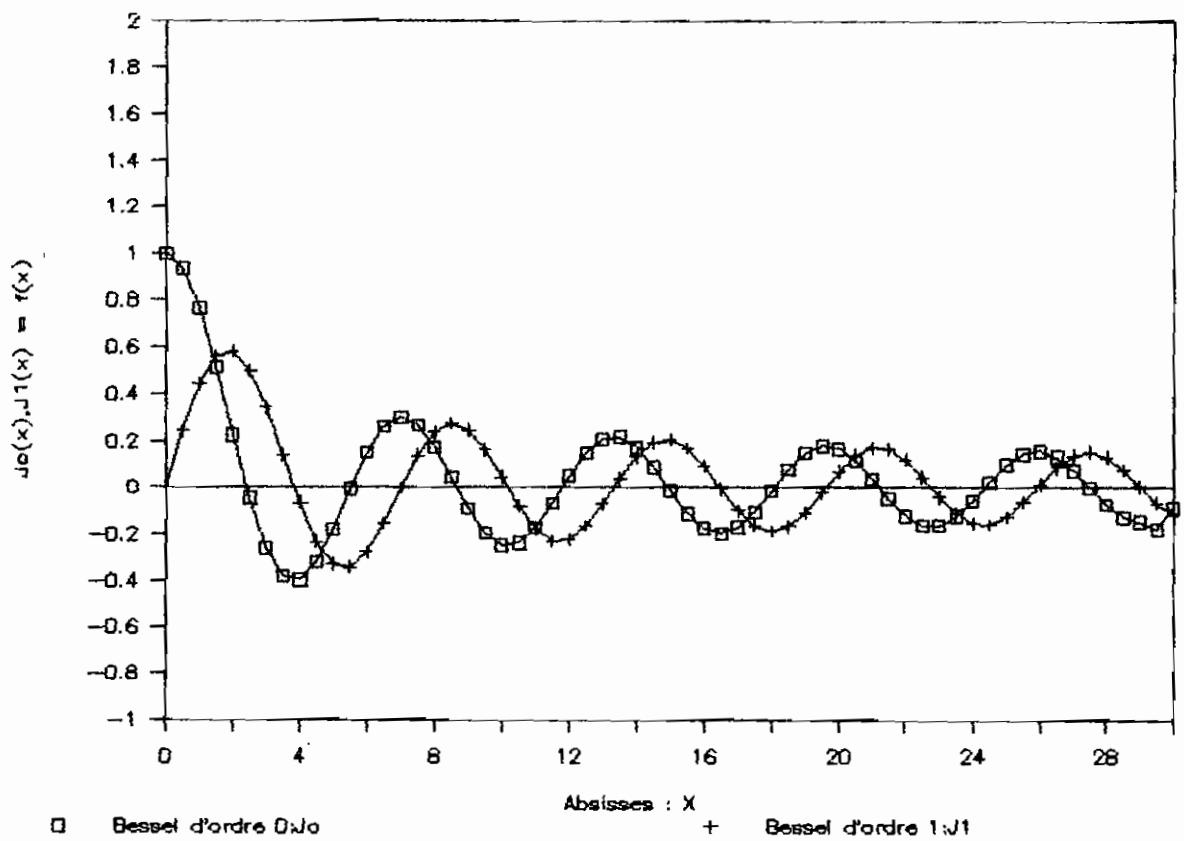
LA BIBLIOGRAPHIE

- [1]: Centre Epérimental de Recherches et d'Etudes du
Bâtiment et des Travaux Publics: CEBTP
Guide Pratique de Dimensionnement des Chaussées pour
les Pays Tropicaux.
(1984)
- [2]: G. Jeuffroy _ R. Sauterey
Cours de Routes. Dimensionnement des Chaussées
Presses de l'école nationale des Ponts et Chaussées
(1984)
- [3]: Maguette DIOP
Cours d'Analyse numérique 1988-1989
Ecole Polytechnique de Thiès: Maguette DIOP.
- [4]: NAI C. Yang
Design of Functional Pavements
MCGRAW-HILL BOOK COMPANY (1972)
- [5]: R. Crottaz, M.L.Pigois
Construction de la Superstructure Routière T.1
(LAVOC)
Ecole Polytechnique Fédérale de Lausanne
(1978)
- [6]: FRANS VAN Cauwelaert
L'Elasticité Anisotrope Appliquée à la Mécanique des
Milieux Granulaires et des Roches
Ecole Polytechnique Fédérale de Lausanne
Thèse N° 477 (1983)
- [7]: William H. Press, Brian P. Flannery,
William T. Vetterling
Numerical Recipes
(USA)
- [8]: J. E. ROMAN
Stresses, Strains and Deflections in Elastic Layered
systems.
Centre de Recherches Routières (CRR _ Bruxelles)
Research report N° 151/JER/1970

ANNEXES

A1- COURBES ET RACINES DES FONCTIONS DE BESSEL J_0 ET J_1

Courbes de Fonctions de Bessel J_0 et J_1

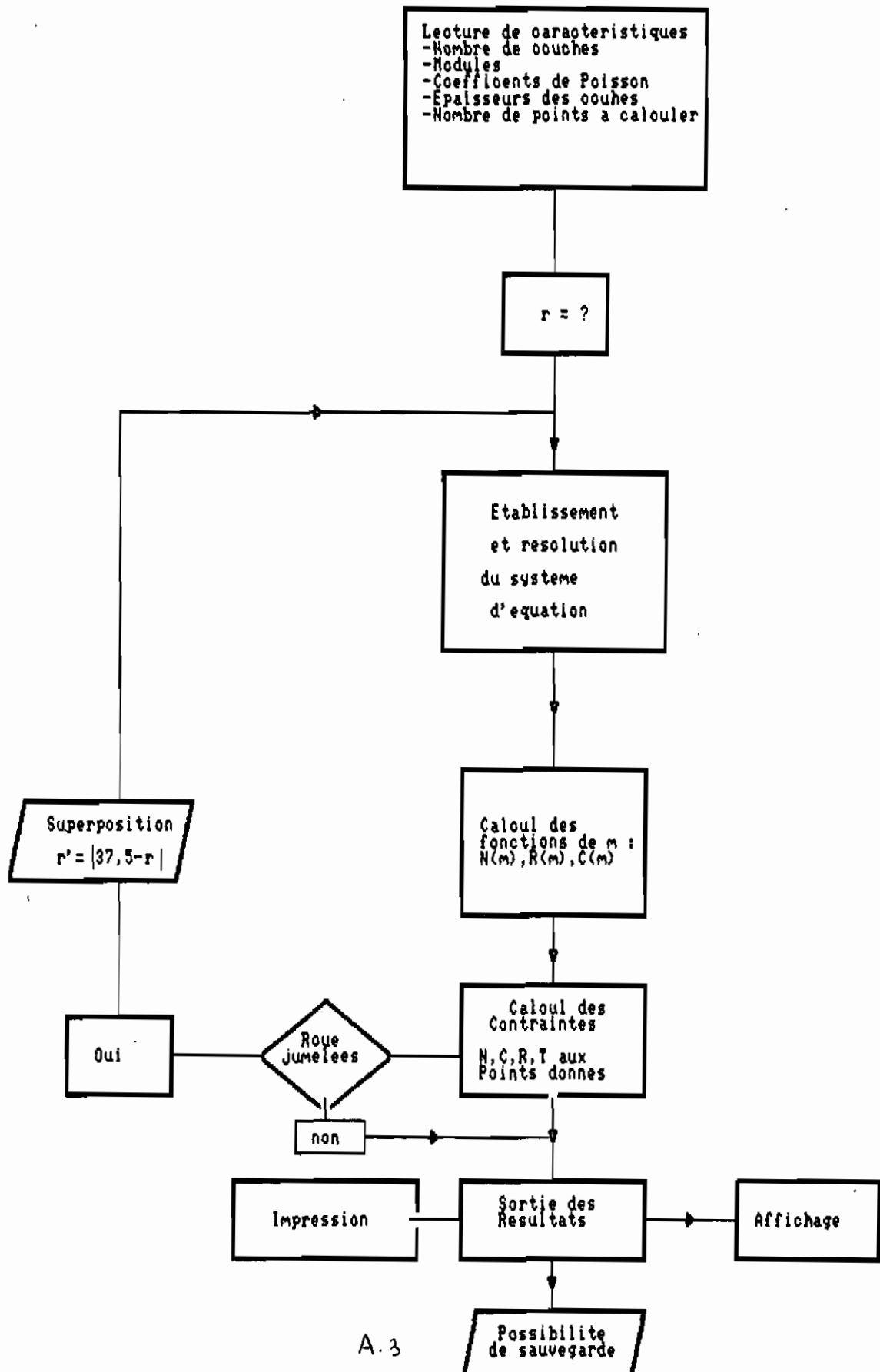


LES RACINES DES FONCTIONS DE BESSEL J_0 ET J_1
DE 0 A 30 OBTENUES PAR DICHOTOMIE AVEC UNE
PRECISION 10^{-8}

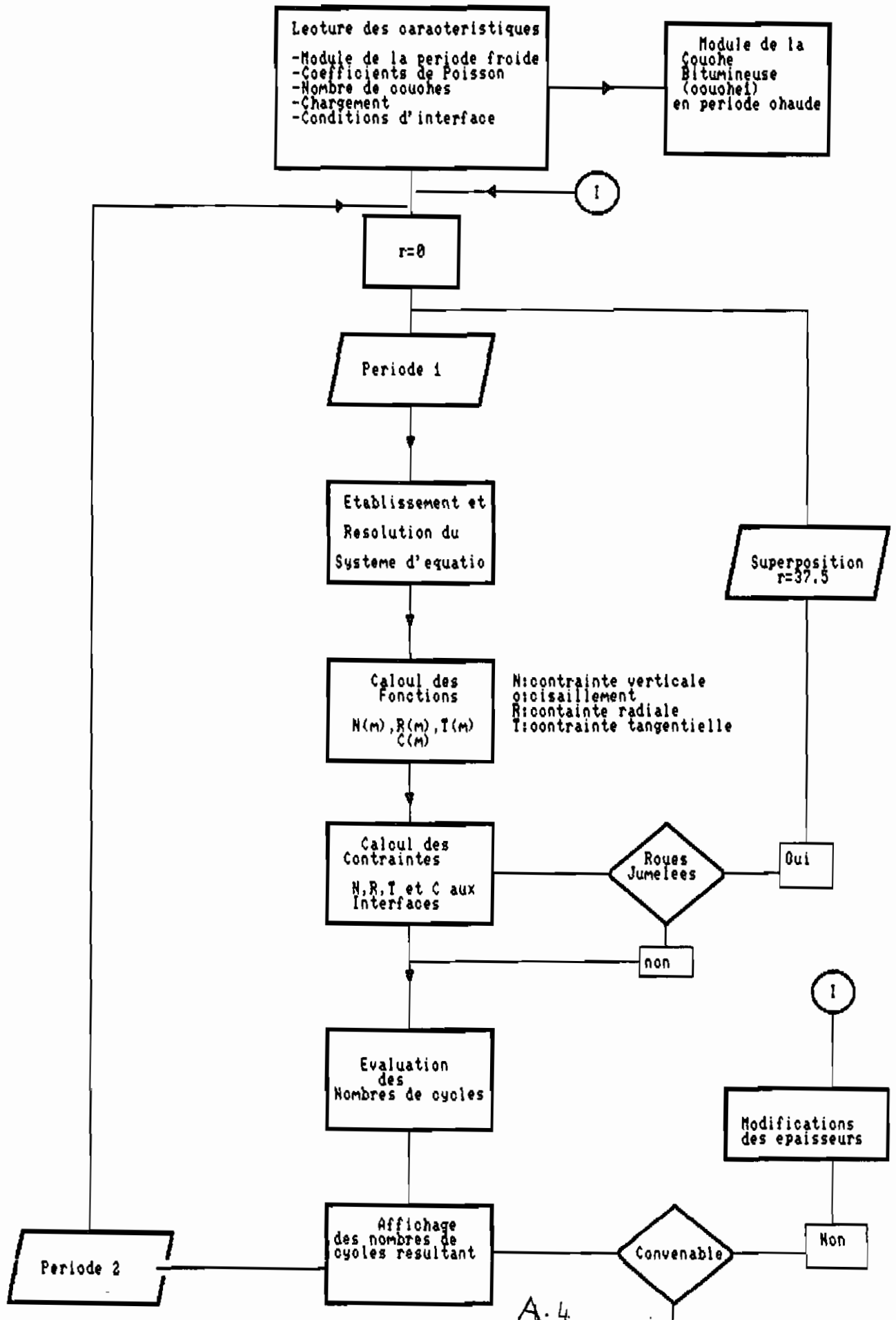
	<u>Pour J_0</u>	<u>Pour J_1</u>
1	2.40483	0.00000
2	5.52008	3.83171
3	8.65373	7.01558
4	11.79153	10.17347
5	14.93092	13.32369
6	18.07106	16.47063
7	21.21164	19.61586
8	24.35247	22.76008
9	27.49348	25.90368
10		29.04683

A2- ORGANIGRAMME DU LOGICIEL ZOVROUTE

1 Calcul de contraintes



II Dimensionnement



A3- LISTING DU PROGRAMME

```

Unit unite2; (Etablissement et résolution du système d'équations)
  Interface
($n+)
uses crt,dos;

Type
  iter=0..6000;
  int=0..25;
  typedonnee=
    Record
      rac1,rac2:array[1..192] of real;
    End;
  tdonnee=
    Record
      t2,tsup2:array[0..5,1..2,1..6] of real;
      t3:array[1..5,1..6] of real;
      t1:array[1..5,1..4] of real;
      condi_int,c_it,nc1,np1:int;
      cycl:array[0..5] of real;
      all:array[1..2] of real;
      cr1,qoo,v1,mol2,r1:real;
      jumelage,ct1:char;
      ope:string;
    End;
  EcranGere=array[1..80,1..25] of integer;
Const
  entier: set of int= [1,2,3,4,5];

Var
  f13,f14,f23,f24,fsup14,fsup13,fsup23,fsup24:array[0..6] of real;
  fd1,fd2,fd3,fd4,fdsup3,fdsup4:array[0..6,1..2] of real;
  zi,f1,f2,f3,f4,fsup3,fsup4:array[0..8] of real;
  a,b,hh1,l,m,mu,mo,cr,vm,qo,r,ta,z,zo,zz:real;
  i,j,k,kl,nbvaieur,n,ordre,p,q,w1,w2:byte;
  itf1,interf,kk,n2,nc,np,incontr:int;
  contrsup,contr:array[1..5] of real;
  rac11,rac12:array[1..192] of real;
  nom_fichier,mau,operation:string;
  ma:array[1..18,1..18] of double; (double precision)
  h,mm,el,w:array[1..5] of real;
  f:array[3..6,1..4] of double; (double precision)
  cyc,cycl:array[0..5] of real;
  md1,a1:array[1..2] of real;
  x,u:array[1..18] of real;
  c,c1,ct,dual:char;
  Ecran:ecrangere;
  ny:real;
  it:iter;

```



```

Function readcar(var touche_spe:boolean):char;
Function bessj0(x:real):real;
Function bessj1(x:real):real;
Procedure lecture;
Procedure localisation;
Procedure systli(m:real);
Procedure cadrel(i1,j1,i2,j2:byte);
Procedure prendre(i1,j1,i2,j2:byte);
Procedure restitue(i1,j1,i2,j2:byte);

```

Implementation

```

Function readcar(var touche_spe:boolean):char;
Var c:char;
Begin
  touche_spe:=false;c:=readkey;
  if c=#0 then
    begin
      touche_spe:=true;
      c:=readkey;
    end;
  readcar:=c
End;

Function bessj0(x:real):real;
Var
  ax,xx,z:real;y,ans,ans1,ans2 :real;
Begin
  If x<8 then
    begin
      y:=sqr(x) ;
      ans1:=1e+10+y*(-2.5e+9+y*(1.5625e+8+y*(-4.3402777e+6)));
      ans1:=ans1+y*y*y*y*(6.781684e+4-6.781684e+2*y);
      ans1:=ans1+y*y*y*y*y*y*(4.7095027+y*(-2.4028075e-2+
      y*(9.3859669e-5)));
      ans1:=ans1+y*y*y*y*y*y*y*y*y*(-2.8969033e-7);
      ans:=ans1+y*y*y*y*y*y*y*y*y*(7.2422584e-10+
      y*(-1.4963343e-12));
      ans:=ans*1e-10;
      bessj0:=ans
    end
  Else begin
      z:=8/x;y:=sqr(z);xx:=x-0.785398164;
      ans1:=1+y*(-1.098628627e-3+y*(2.734510407e-5+
      y*(-2.073370639e-6)));
      ans1:=ans1+y*y*y*y*2.093887211e-7;
      ans2:=-1.562499995e-2+y*(1.430488765e-4-y*6.911147651e-6);
      ans2:=ans2+y*y*y*(7.621095161e-7-y*9.34945152e-8);
      ans:= sqrt(0.636619772/x)*(cos(xx)*ans1-z*sin(xx)*ans2);
      bessj0:=ans
    end
  End; ( Calcul J0(x))

```

```

Function bessj1(x:real):real;
Var
  ax,xx,z:real;y,ans,ans1,ans2 : real;
Begin
  If x<8 then
    begin
      y:=sqr(x);ans1:=x*(72362614.232e+3+y*(-7895059.235e+3+
      y*242396853.1));
      ans1:=ans1+x*y*y*y*(-2972611.439+y*(15704.4826-
      -30.16036606*y));
      ans2:=144725228.442e+3+y*(2300535.178e+3+y*(18583304.74+
      y*99447.43394));
      ans2:=ans2+y*y*y*y*(376.9991397+y);
      bessj1:=ans1/ans2
    end
  Else begin
      z:=8/x;y:=sqr(z);xx:=x-2.356194491;
      ans1:=1+y*(1.83105e-3+y*(-3.516396496e-5));
      ans1:=ans1+y*y*y*(2.457520174e-6+y*(-2.40337019e-7));
      ans2:=0.04687499995+y*(-2.002690873e-4+
      y*(8.449199096e-6));
      ans2:=ans2+y*y*y*(-8.8228987e-7+1.05787412e-7*y);
      ans:= sqrt(0.636619772/x)*(cos(xx)*ans1-z*sin(xx)*ans2);
      bessj1:=ans;
    end
  End;      (Fonction J1)

Procedure cadre1(i1,j1,i2,j2:byte);
var
  i:byte;
Begin
  gotoxy(j1,i1);write(chr(218));
  for i:=j1+1 to j2-1 do
    Begin
      gotoxy(i,i1);write(chr(196));
      gotoxy(i,i2);write(chr(196));
    End;
  gotoxy(j2,i1);writeln(chr(191));
  gotoxy(j1,i2);write(chr(192));
  for i:=i1+1 to i2-1 do
    begin
      gotoxy(j1,i);write(chr(179));
      gotoxy(j2,i);write(chr(179));
    end;
  gotoxy(j2,i2);writeln(chr(217));
End;

```

```

Procedure Getscreen(x1,y1,x2,y2:byte; var recup:ecrangere );
  var
    x,y:integer;mot:word;
    etat:registers;
  Begin
    etat.ax:=15*256;
    intr(16,etat);
    if Lo(etat.ax) in [2,7] then mot:=$B000 Else mot:=$B800;
    for y:=y1 to y2 do
      for x:=x1 to x2 do
        recup[x,y]:=MemW[mot:(x-1)*2+(y-1)*160]
      End;
    End;

```

```

Procedure Putscreen(x1,y1,x2,y2:byte; var recup:ecrangere );
  var
    x,y:integer;mot:word;
    etat:registers;
  Begin
    etat.ax:=15*256;
    intr(16,etat);
    if Lo(etat.ax) in [2,7] then mot:=$B000 Else mot:=$B800;
    for y:=y1 to y2 do
      for x:=x1 to x2 do
        MemW[mot:(x-1)*2+(y-1)*160]:=recup[x,y]
      End;
    End;

```

```

Procedure prendre(i1,j1,i2,j2:byte);
  Begin
    Getscreen(j1,i1,j1+j2,i1+i2,ecran);
    cadrel(i1,j1,i1+i2,j1+j2);
    window(j1+1,i1+1,j1+j2-1,i1+i2-1);clrscr;
    window(1,1,80,25);
  End;

```

```

Procedure restitue(i1,j1,i2,j2:byte);
  Begin
    putscreen(j1,i1,j1+j2,i1+i2,ecran);
  End;

```

```

Procedure chgt;
  Begin
    w2:=(j+3) div 4;mu:=el[w1]*(1+mm[w2])/el[w2];
    p:=i-4*w1+4;q:=j-4*w2+4;
    ordre:=10*p+q
  End;

```

```

Procedure localisation;
  var
    hh2 :real;
    bs:boolean;
  Begin
    n2:=1;hh1:=0;hh2:=h[1];bs:=false;

```

```

repeat
  If nc>1 then
    begin
      if (hh1<=z)and(hh2>z) then
        begin
          bs:=true;
        end
      else begin
        hh1:=hh1+h[n2];hh2:=hh1+h[n2+1];
        n2:=n2+1;
        end;
      if (hh1<=z)and(n2>=nc) then
        begin
          n2:=nc;bs:=true
        end;
      end
    Else begin
      n2:=1;bs:=true
    end;
  until bs;
End;

Procedure lecture;
Var
  fichierdonnee:file of typedonnee;
      donnee:      typedonnee;
  nomfichier:      string[12];

Begin
  nbvaleur:=trunc(1/3.13);

  nomfichier:='a:\victor1';
  assign(fichierdonnee,nomfichier);
  reset(fichierdonnee);
  read(fichierdonnee,donnee);
  with donnee do
    begin
      for i:=1 to nbvaleur do
        begin
          rac1[i]:=rac1[i];
          rac12[i]:=rac2[i+1]
        end
      end;
  close(fichierdonnee)
End;

Procedure choix(m:real);
Begin
  Case ordre of
    31: f[3,1]:=-m*exp(m*zz);
    32: f[3,2]:=(1-2*mm[w2]-m*zz)*exp(m*zz);
    33: f[3,3]:=m*exp(-m*zz);
  end

```

```

34: f[3,4]:=(1-2*mm[w2]+m*zz)*exp(-m*zz);
41: f[4,1]:=m*exp(m*zz);
42: f[4,2]:=(2*mm[w2]+m*zz)*exp(m*zz);
43: f[4,3]:=m*exp(-m*zz);
44: f[4,4]:=(m*zz-2*mm[w2])*exp(-m*zz);
51: f[5,1]:=mu*m*exp(m*zz);
52: f[5,2]:=mu*(m*zz+1)*exp(m*zz);
53: f[5,3]:=-mu*m*exp(-m*zz);
54: f[5,4]:=mu*(1-m*zz)*exp(-m*zz);
61: f[6,1]:=-mu*m*exp(m*zz);
62: f[6,2]:=mu*(2-4*mm[w2]-m*zz)*exp(m*zz);
63: f[6,3]:=-mu*m*exp(-m*zz);
64: f[6,4]:=mu*(4*mm[w2]-m*zz-2)*exp(-m*zz)
End ; { Fin case)
End;

```

```

Procedure matrice(m:real);

```

```

var ci:int;

```

```

Begin

```

```

n:=4*nc-2;

```

```

If nc=1 then

```

```

Begin

```

```

ma[1,1]:=m*m*m;ma[1,2]:=m*m*(1-2*mm[1]);

```

```

ma[2,1]:=m;ma[2,2]:=-2*mm[1]

```

```

End

```

```

Else begin

```

```

ma[1,1]:=-m*m*m;ma[1,2]:=m*m*(1-2*mm[1]);

```

```

ma[1,3]:=m*m*m;

```

```

ma[1,4]:=m*m*(1-2*mm[1]);ma[2,1]:=m;ma[2,2]:=2*mm[1];

```

```

ma[2,3]:=m;ma[2,4]:=-2*mm[1];

```

```

for i:=1 to 2 do

```

```

begin

```

```

for j:=5 to n do

```

```

ma[i,j]:=0

```

```

end

```

```

end;

```

```

Case interf of

```

```

0: Begin

```

```

For i:=3 to n do

```

```

Begin

```

```

w1:=(i+1) div 4;zz:=0;

```

```

for ci:=1 to w1 do zz:=zz+h[ci];

```

```

For j:=1 to 4*w1-4 do

```

```

ma[i,j]:=0;

```

```

For j:=4*w1-3 to 4*w1 do

```

```

Begin

```

```

chgt;

```

```

choix(m);

```

```

ma[i,j]:=f[p,q]

```

```

End;

```

```

If 4*w1+1<n-1 then
  Begin
    For j:=4*w1+1 to 4*w1+4 do
      Begin
        chgt;
        choix(m);
        ma[i,j]:=-f[p,q]
      End
    End
  End
Else begin
  for j:=4*w1+1 to n do
    begin
      w2:=(j+3) div 4;
      mu:=el[w1]*(1+mm[w2])/el[w2];
      p:=i-4*w1+4;q:=j-4*w2+6;
      ordre:=10*p+q;
      choix(m);
      ma[i,j]:=-f[p,q]
    end
  end;

  For j:=4*w1+5 to n do
    ma[i,j]:=0
  End {Boucle sur i}
End; {Case de 0}

1: Begin
  For i:=3 to n do
    Begin
      w1:=(i+1) div 4;zz:=0;
      for ci:=1 to w1 do zz:=zz+h[ci];
      For j:=1 to 4*w1-4 do
        ma[i,j]:=0;

      For j:=4*w1-3 to 4*w1 do
        Begin
          chgt;
          choix(m);
          ma[i,j]:=f[p,q];
          If itf1=12 then
            Begin
              If i=5 then ma[i,j]:=0
            End
          Else if itf1=23 then
            begin
              if i=9 then ma[i,j]:=0
            end
          End;
        End
      If 4*w1+1<n-1 then
        Begin
          For j:=4*w1+1 to 4*w1+4 do
            Begin

```

```

    If itf1=12 then
      begin
        if i=4 then ma[i,j]:=0;
        if i=5 then
          begin
            w2:=(j+3) div 4;q:=j-4*w2+4;
            mu:=el[w1]*(1+mm[w2])/el[w2];
            ordre:=40+q;
            choix(m);
            ma[i,j]:=f[4,q]
          end
        else if i<>4 then
          begin
            chgt;
            choix(m);
            ma[i,j]:=-f[p,q];
          end
        end
      Else if itf1=23 then
        begin
          if i=8 then ma[i,j]:=0;
          if i=9 then
            begin
              w2:=(j+3) div 4;q:=j-4*w2+4;
              mu:=el[w1]*(1+mm[w2])/el[w2];
              ordre:=40+q;
              choix(m);
              ma[i,j]:=f[4,q]
            end
          else if i<>8 then
            begin
              chgt;
              choix(m);
              ma[i,j]:=-f[p,q];
            end
          end
        end
      End
    End
  Else Begin
    For j:=4*w1+1 to n do
      Begin
        w2:=(j+3) div 4;
        mu:=el[w1]*(1+mm[w2])/el[w2];
        p:=i-4*w1+4;q:=j-4*w2+6;
        ordre:=10*p+q;
        If itf1=12 then
          begin
            if i=4 then ma[i,j]:=0;
            if i=5 then
              begin
                ordre:=40+q;
                choix(m);
              end
            end
          end
        end
      end
    end
  end

```

```

        ma[i,j]:=f[4,q]
    end
    else if i<>4 then
        begin
            choix(m);
            ma[i,j]:=-f[p,q];
        end
    end
    Else if itf1=23 then
        begin
            if i=8 then ma[i,j]:=0;
            if i=9 then
                begin
                    ordre:=40+q;
                    choix(m);
                    ma[i,j]:=f[4,q]
                end
            else if i<>8 then
                begin
                    choix(m);
                    ma[i,j]:=-f[p,q]
                end
            end
        end
    end
    End
End;

    For j:=4*w1+5 to n do
        ma[i,j]:=0
    End {Boucle sur i}
End;      {Case de 1}

2: Begin
    For i:=3 to n do
        Begin
            w1:=(i+1) div 4;zz:=0;
            for ci:=1 to w1 do zz:=zz+h[ci];
            For j:=1 to 4*w1-4 do
                ma[i,j]:=0;
            For j:=4*w1-3 to 4*w1 do
                Begin
                    chgt;
                    choix(m);
                    ma[i,j]:=f[p,q];
                    If i=5 then ma[i,j]:=0;
                    If i=9 then ma[i,j]:=0
                End;
            If 4*w1+1<n-1 then
                Begin
                    For j:=4*w1+1 to 4*w1+4 do
                        Begin
                            chgt;

```



```

choix(m);
ma[i,j]:=-f[p,q];
If i=4 then ma[i,j]:=0;
If i=5 then
  Begin
    w2:=(j+3) div 4;q:=j-4*w2+4;
    mu:=el[w1]*(1+mm[w2])/el[w2];
    ordre:=40+q;
    choix(m);
    ma[i,j]:=f[4,q]
  End;
If i=8 then ma[i,j]:=0;
If i=9 then
  Begin
    w2:=(j+3) div 4;q:=j-4*w2+4;
    mu:=el[w1]*(1+mm[w2])/el[w2];
    ordre:=40+q;
    choix(m);
    ma[i,j]:=f[4,q]
  End;
End
End
Else Begin
  For j:=4*w1+1 to n do
    Begin
      w2:=(j+3) div 4;
      mu:=el[w1]*(1+mm[w2])/el[w2];
      p:=i-4*w1+4;q:=j-4*w2+6;
      ordre:=10*p+q;
      choix(m);
      ma[i,j]:=-f[p,q];
      If i=4 then ma[i,j]:=0;
      If i=5 then
        Begin
          ordre:=40+q;
          choix(m);
          ma[i,j]:=f[4,q]
        End;
      If i=8 then ma[i,j]:=0;
      If i=9 then
        Begin
          ordre:=40+q;
          choix(m);
          ma[i,j]:=f[4,q]
        End
      End
    End
  End;
  For j:=4*w1+5 to n do
    ma[i,j]:=0
  End (Boucle sur i)
End;      (Case de 2)

```

```

    End      (Fin case)
  End;      (Fin procedurematrice)

Procedure Elimination;
Var
  b1,con,e:double; {double precision}
  ul:real;

Begin
  For k:=1 to n-1 do
    Begin
      w1:=(k+3) div 4;
      If 4*w1<n-4 then
        Begin
          For i:=k+1 to 4*w1+2 do
            Begin
              If abs(ma[k,k])<abs(ma[i,k]) then
                Begin
                  b1:=ma[k,k];ma[k,k]:=ma[i,K];ma[i,k]:=b1;
                  For j:=k+1 to 4*w1+4 do
                    Begin
                      con:=ma[k,j];ma[k,j]:=ma[i,j];
                      ma[i,j]:=con;
                    End;
                  ul:=u[k];u[k]:=u[i];u[i]:=ul;
                End
              End
            End
          End
        Else Begin
          For i:=k+1 to n do
            Begin
              If abs(ma[k,k])<abs(ma[i,k]) then
                Begin
                  b1:=ma[k,k];ma[k,k]:=ma[i,K];ma[i,k]:=b1;
                  For j:=k+1 to n do
                    Begin
                      con:=ma[k,j];ma[k,j]:=ma[i,j];
                      ma[i,j]:=con;
                    End;
                  ul:=u[k];u[k]:=u[i];u[i]:=ul;
                End
              End
            End
          End;
        If 4*w1<n-4 then
          Begin
            For i:=k+1 to 4*w1+2 do
              Begin
                e:=ma[i,k];
                for j:=k to 4*w1+4 do
                  Begin
                    if (abs(e)>10)and(abs(ma[k,j])>10)then
                      ma[i,j]:=ma[i,j]-ma[k,j]*(e/ma[k,k])
                  End
                End
              End
            End
          End
        End
      End;
    End
  End

```

```

        else ma[i,j]:=ma[i,j]-(ma[k,j]*e)/ma[k,k];
      End;
      u[i]:=u[i]-u[k]*e/ma[k,k];
    End
  End
Else Begin
  For i:=k+1 to n do
    Begin
      e:=ma[i,k];
      for j:=k to n do
        Begin
          if (abs(e)>10)and(abs(ma[k,j])>10) then
            ma[i,j]:=ma[i,j]-ma[k,j]*(e/ma[k,k])
          elsema[i,j]:=ma[i,j]-(ma[k,j]*e)/ma[k,k];
        End;
        u[i]:=u[i]-u[k]*e/ma[k,k]
      End
    End
  End;
End; (Fin de la ProcedureElimination)

```

Procedure solution;

Var

som:real;

Begin

(La remontee triangulaire)

x[n]:=u[n]/ma[n,n];

If n>=6 then

Begin

For i:=n-1 downto n-5 do

Begin

som:=0;

For j:=n downto i+1 do

Begin

som:=som+ma[i,j]*x[j];

End;

x[i]:=(u[i]-som)/ma[i,i];

End;

For i:=n-6 downto 1 do

Begin

som:=0;w1:=(n-i-2) div 4;

for j:=n-4*w1+2 downto i+1 do

som:=som+ma[i,j]*x[j];

x[i]:=(u[i]-som)/ma[i,i];

End

End

Else begin

For i:=n-1 downto 1 do

Begin

som:=0;

For j:=n downto i+1 do

```

        som:=som+ma[i,j]*x[j];
        x[i]:=(u[i]-som)/ma[i,i]
    End
    end;
End; {fin}

Procedure systli(m:real);
Begin
    u[1]:=-1;
    for i:= 2 to 4*nc-2 do
        u[i]:=0;
        matrice(m);
        elimination;
        solution;
    End;    { Fin de la procedure systli}
End.

```

```

Unit unite3;{Intégration des fonctions}
  Interface
uses crt,unite2;

  Procedure integration;

    Implementation
  Procedure calculz(m:real);
var
  b1,z1:real;
  ev,id:int;
  Procedure p1;
  Begin
    case itf1 of
      12: begin
        If r<>0 then
          Begin
            if (nc=2)and(z=h[1]) then
              begin
                contr[2]:=0;
                contrsup[3]:=-b1*((1-m*h[1])*x[6]-m*x[5])
                  *exp(-m*h[1])/(m*r)-1e-15;
                contrsup[4]:=-contrsup[3]
              end
            else if (nc>2)and(z=h[1]) then
              begin
                contr[2]:=0;
                contrsup[3]:=-b1*(m*x[5]+(1+m*z)*x[6])
                  *exp(m*z)/(m*r)-1e-15;
                contrsup[3]:=contrsup[3]-b1*((1-m*z)*x[8]-m*x[7])
                  *exp(-m*z)/(m*r);
                contrsup[4]:=-contrsup[3]
              end;
            End
          Else
            Begin
              if (nc=2)and(z=h[1]) then
                begin
                  contr[2]:=0;
                  contrsup[3]:=-6.25*m*m*bessj1(12.5*m)*
                    ((1-m*h[1])*x[6]-m*x[5])*exp(-m*h[1])-1e-15;
                  contrsup[4]:=-contrsup[3]
                end
              else if (nc>2)and(z=h[1]) then
                begin
                  contr[2]:=0;
                  contrsup[3]:=-(m*x[5]+(1+m*z)*x[6])*exp(m*z)

```

```

        -((1-m*z)*x[8]-m*x[7])*exp(-m*z)-1e-15;
contrsup[3]:=contrsup[3]*6.25*m*m*bessj1(12.5*m);
contrsup[4]:=-contrsup[3]
    end;
    End;
end;
23: begin
    if (nc=3)and(z=h[1]+h[2]) then
    begin
        contr[2]:=0;
        if r<>0 then
        begin
            contrsup[3]:=-b1*((1-m*z)*x[10]-m*x[9])
                *exp(-m*z)/(m*r)-1e-15;
            contrsup[4]:=-contrsup[3]
                end
            else
            begin
                contrsup[3]:=-6.25*m*m*bessj1(12.5*m)
                    *((1-m*z)*x[10]-m*x[9])*exp(-m*z)-1e-15;
                contrsup[4]:=-contrsup[3]
                    end
                end
            else if (nc>3)and(z=h[1]+h[2]) then
            begin
                contr[2]:=0;
                if r<>0 then
                begin
                    contrsup[3]:=-b1*(m*x[9]+(1+m*z)*x[10])
                        *exp(m*z)/(m*r)-1e-15;
                    contrsup[3]:=contrsup[3]-b1*((1-m*z)*x[12]-m*x[11])
                        *exp(-m*z)/(m*r);
                    contrsup[4]:=-contrsup[3]
                        end
                    else
                    begin
                        contrsup[3]:=-(m*x[9]+(1+m*z)*x[10])*exp(m*z)-
                            ((1-m*z)*x[12]-m*x[11])*exp(-m*z)-1e-15;
                        contrsup[3]:=contrsup[3]*6.25*m*m*bessj1(12.5*m);
                        contrsup[4]:=-contrsup[3]
                            end
                            end
                        end;
                    end {case itf1}
                End;
Procedure p2;
Begin
    if r<>0 then
    begin
        if ((z=h[1])and(nc=2))or((z=h[1]+h[2])and(nc=3)) then
        begin

```

```

    ev:=4*n2-3;contr[2]:=0;
    contrsup[3]:=-b1*((1-m*z)*x[ev+1]-m*x[ev])
                *exp(-m*z)/(m*r)-1e-15;
    contrsup[4]:=-contrsup[3]
end
else
  if ((z=h[1])and(nc>2))or((z=h[1]+h[2])and(nc>3)) then
  begin
    ev:=4*n2-7;contr[2]:=0;
    contrsup[3]:=-b1*(m*x[ev]+(1+m*z)*x[ev+1])
                *exp(m*z)/(m*r)-1e-15;
    contrsup[3]:=contrsup[3]-b1*((1-m*z)*x[ev+3]-m*x[ev+2])
                *exp(-m*z)/(m*r);
    contrsup[4]:=-contrsup[3]
  end;
end
else
  begin
    if ((z=h[1])and(nc=2))or((z=h[1]+h[2])and(nc=3)) then
    begin
      ev:=4*n2-3;contr[2]:=0;
      contrsup[3]:=-6.25*m*m*bessj1(12.5*m)
      *(((1-m*z)*x[ev+1]-m*x[ev])*exp(-m*z))-1e-15;
      contrsup[4]:=-contrsup[3]
    end
    else
    begin
      if ((z=h[1])and(nc>2))or((z=h[1]+h[2])and(nc>3)) then
      begin
        ev:=4*n2-7;contr[2]:=0;
        contrsup[3]:=-(m*x[ev]+(1+m*z)*x[ev+1])
                    *exp(m*z)-1e-15;
        contrsup[3]:=contrsup[3]-((1-m*z)*x[ev+3]+m*x[ev+2])
                    *exp(-m*z);
        contrsup[3]:=contrsup[3]*6.25*m*m*bessj1(12.5*m);
        contrsup[4]:=-contrsup[3]
      end;
    end;
  end
end;
End;

Begin
  localisation;
  systli(m);
  Case incontr of

  1: Begin
    if r=0 then b1:=12.5*m*m*bessj1(12.5*m)
    else b1:=12.5*m*m*bessj0(m*r)*bessj1(12.5*m);
    If z=0 then
    Begin
      contr[1]:=b1*(-m*x[1]+(1-2*mm[1])*x[2]+

```

```

            m*x[3]+(1-2*mm[1])*x[4]);
    contr[3]:=b1*(m*x[1]+(1+2*mm[1])*x[2]-m*x[3]+
            (1+2*mm[1])*x[4]);
    contr[4]:=b1*2*mm[1]*(x[2]+x[4]);
End;
If n2=nc then
  Begin
    If z<>hh1 then
      Begin
        ev:=4*n2-3;

contr[1]:=b1*(m*x[ev]+(1-2*mm[nc]+m*z)*x[ev+1])*exp(-m*z)-1e-15;
contr[3]:=b1*(-m*x[ev]+(1+2*mm[nc]-m*z)*x[ev+1])*exp(-m*z)+1e-15;
      contr[4]:=b1*2*mm[nc]*x[ev+1]*exp(-m*z)+1e-15;
      End
    Else if nc>1 then
      begin
        ev:=4*nc-7;

contr[1]:=(-m*x[ev]+(1-2*mm[nc-1]-m*z)*x[ev+1])*exp(m*z)-1e-15;
contr[1]:=contr[1]+(m*x[ev+2]+m*z*x[ev+3])*exp(-m*z);
contr[1]:=b1*(contr[1]+(1-2*mm[nc-1])*x[ev+3]*exp(-m*z))-1e-15;
contr[3]:=(m*x[ev]+(1+2*mm[nc-1]+m*z)*x[ev+1])*exp(m*z)+1e-15;
      contr[3]:=contr[3]-m*x[ev+2]*exp(-m*z)+1e-15;

contr[3]:=contr[3]+(1+2*mm[nc-1])*exp(-m*z)*x[ev+3];
      contr[3]:=b1*(contr[3]-m*z*exp(-m*z)*x[ev+3]);
      contr[4]:=b1*x[ev+1]*2*mm[nc-1]*exp(m*z)+1e-15;

contr[4]:=contr[4]+b1*2*mm[nc-1]*x[ev+3]*exp(-m*z);
      end
    End
  Else
    Begin
      If z<>hh1 then
        Begin
          ev:=4*n2-3;

contr[1]:=(-m*x[ev]+(1-2*mm[n2]-m*z)*x[ev+1])*exp(m*z)-1e-15;
          contr[1]:=contr[1]+m*x[ev+2]*exp(-m*z);
          contr[1]:=contr[1]+(1-2*mm[n2])*x[ev+3]*exp(-m*z);
          contr[1]:=b1*(contr[1]+m*z*x[ev+3]*exp(-m*z));

contr[3]:=(m*x[ev]+(1+2*mm[n2]+m*z)*x[ev+1])*exp(m*z)+1e-15;

contr[3]:=contr[3]+(-m*x[ev+2]+(1+2*mm[n2])*x[ev+3])*exp(-m*z);
          contr[3]:=b1*(contr[3]-m*z*exp(-m*z)*x[ev+3]);
          contr[4]:=b1*x[ev+1]*2*mm[n2]*exp(m*z)+1e-15;
        End
      End
    End
  End
End

```



```

        contr[4]:=contr[4]+b1*2*mm[n2]*x[ev+3]*exp(-m*z);
        End
    Else if n2>1 then
        Begin
            ev:=4*n2-7;

contr[1]:=(-m*x[ev]+(1-2*mm[n2-1]-m*z)*x[ev+1])*exp(m*z)-1e-15;
            contr[1]:=contr[1]+m*x[ev+2]*exp(-m*z);

contr[1]:=contr[1]+(1-2*mm[n2-1])*x[ev+3]*exp(-m*z);
            contr[1]:=b1*(contr[1]+m*z*x[ev+3]*exp(-m*z));

contr[3]:=(m*x[ev]+(1+2*mm[n2-1]+m*z)*x[ev+1])*exp(m*z)+1e-15;
contr[3]:=contr[3]-(m*x[ev+2]+m*z*x[ev+3])*exp(-m*z);

contr[3]:=b1*(contr[3]+(1+2*mm[n2-1])*exp(-m*z)*x[ev+3]);
            contr[4]:=b1*2*mm[n2-1]*x[ev+1]*exp(m*z)+1e-15;

contr[4]:=contr[4]+b1*2*mm[n2-1]*x[ev+3]*exp(-m*z);
        End
    End;
    If nc>=2 then
        Begin
            case interf of
            1: begin
                    case itf1 of
                    12: begin
                            if (z=h[1])and(nc>2) then
                                begin
                                    contrsup[3]:=(1+2*mm[2]+m*h[1])*exp(m*h[1])*x[6]-1e-15;
contrsup[3]:=contrsup[3]+m*exp(m*h[1])*x[5]-m*x[7]*exp(-m*h[1]);
contrsup[3]:=b1*(contrsup[3]+(1+2*mm[2]-m*h[1])*exp(-m*h[1])*x[8]
);
contrsup[4]:=2*mm[2]*b1*(x[6]*exp(m*h[1])+x[8]*exp(-m*h[1]))-1e-1
5;

                                end
                            else if (z=h[1])and(nc=2) then
                                begin

contrsup[3]:=b1*(-m*x[5]+(1+2*mm[2]-m*h[1])*x[6])*exp(-m*h[1])-1e
-15;
                                    contrsup[4]:=2*mm[2]*b1*x[6]*exp(-m*h[1]);
                                end
                            end;
                    23: begin
                            if (z=h[1]+h[2])and(nc>3) then
                                begin
                                    contrsup[3]:=(1+2*mm[3]+m*z)*exp(m*z)*x[10]-1e-15;

```

```

contrsup[3]:=contrsup[3]+m*exp(m*z)*x[9]-m*x[11]*exp(-m*z);
contrsup[3]:=b1*(contrsup[3]+(1+2*mm[3]-m*z)*exp(-m*z)*x[12]);
contrsup[4]:=2*mm[3]*b1*(x[10]*exp(m*z)+x[12]*exp(-m*z))-1e-15;
      end
      else if (z=h[1]+h[2])and(nc=3) then
      begin
contrsup[3]:=b1*(-m*x[9]+(1+2*mm[3]-m*z)*x[10])*exp(-m*z)-1e-15;
contrsup[4]:=2*mm[3]*b1*x[10]*exp(-m*z)-1e-15;
      end
      end
      end      (case itf1)
      end;
      2: begin
      if ((z=h[1])and(nc>2))or((z=h[1]+h[2])and(nc>3)) then
      begin
      ev:=4*n2-7;
      contrsup[3]:=(1+2*mm[n2]+m*z)*exp(m*z)*x[ev+1]-1e-15;
contrsup[3]:=contrsup[3]+m*exp(m*z)*x[ev]-m*x[ev+2]*exp(-m*z);
contrsup[3]:=b1*(contrsup[3]+(1+2*mm[n2]-m*z)*exp(-m*z)*x[ev+3]);
contrsup[4]:=2*mm[n2]*b1*(x[ev+1]*exp(m*z)+x[ev+3]*exp(-m*z))-1e-15;
      end
      e l s e i f
      ((z=h[1])and(nc=2))or((z=h[1]+h[2])and(nc=3)) then
      begin
      ev:=4*n2-3;
contrsup[3]:=b1*(-m*x[ev]+(1+2*mm[nc]-m*z)*x[ev+1])*exp(-m*z)-1e-15;
      contrsup[4]:=2*mm[n2]*b1*x[ev+1]*exp(-m*z)-1e-15;
      end
      end
      End;      (case interf)
      End
      End;
      2: Begin
      b1:=12.5*m*m*bessj1(m*r)*bessj1(12.5*m);
      If z=0 then
      Begin
      if r<>0 then
      begin
contr[2]:=b1*(m*x[1]+2*mm[1]*x[2]+m*x[3]-2*mm[1]*x[4])-1e-15;

```

```

        contr[3]:=-b1*(m*x[1]+x[2]+x[4]-m*x[3])/(m*r)+1e-15;
        contr[4]:=b1*(m*x[1]+x[2]+x[4]-m*x[3])/(m*r)+1e-15
    end
    else begin
        contr[2]:=0;

contr[3]:=-6.25*m*m*bessj1(12.5*m)*(m*x[1]+x[2]+x[4]-m*x[3])+1e-15;
        contr[4]:=-contr[3];
        end
    End;
    If n2=nc then
        Begin
            If z<>hh1 then
                Begin
                    ev:=4*n2-3;
                    if r<>0 then
                        begin

contr[2]:=b1*(m*x[ev]+(m*z-2*mm[nc])*x[ev+1])*exp(-m*z)-1e-15;
contr[3]:=-b1*(-m*x[ev]+(1-m*z)*x[ev+1])*exp(-m*z)/(m*r)+1e-15;
contr[4]:=b1*(-m*x[ev]+(1-m*z)*x[ev+1])*exp(-m*z)/(m*r)+1e-15;
                        end
                        else
                            begin
                                contr[2]:=0;

contr[3]:=(-m*x[ev]+(1-m*z)*x[ev+1])*exp(-m*z)-1e-15;
contr[3]:=-6.25*m*m*bessj1(12.5*m)*contr[3]+1e-15;
contr[4]:=(-m*x[ev]+(1-m*z)*x[ev+1])*exp(-m*z)-1e-15;
contr[4]:=6.25*m*m*bessj1(12.5*m)*contr[4]+1e-15
                            end
                            End
                            Else if nc>1 then
                                Begin
                                    ev:=4*nc-7;
                                    if r<>0 then
                                        begin

contr[2]:=(m*x[ev]+(2*mm[nc-1]+m*z)*x[ev+1])*exp(m*z)+1e-15;
                                    contr[2]:=contr[2]+m*exp(-m*z)*x[ev+2];

contr[2]:=b1*(contr[2]+(m*z-2*mm[nc-1])*exp(-m*z)*x[ev+3])-1e-15;
contr[3]:=-b1*(m*x[ev]+(1+m*z)*x[ev+1])*exp(m*z)/(m*r)+1e-15;
contr[3]:=contr[3]-b1*((1-m*z)*x[ev+3]-m*x[ev+2])*exp(-m*z)/(m*r)
                                        end
                                    end
                                end
                            end
                    end
                end
            end
        end
    end

```

```

;
contr[4]:=b1*(m*x[ev]+(1+m*z)*x[ev+1])*exp(m*z)/(m*r)+1e-15;
contr[4]:=contr[4]+b1*((1-m*z)*x[ev+3]-m*x[ev+2])*exp(-m*z)/(m*r)
;
      end
      else
      begin
      contr[2]:=0;
      contr[3]:=-(m*x[ev]+(1+m*z)*x[ev+1])*exp(m*z)+1e-15;
contr[3]:=contr[3]-((1-m*z)*x[ev+3]-m*x[ev+2])*exp(-m*z);
      contr[3]:=contr[3]*6.25*m*m*bessj1(12.5*m);
      contr[4]:=(m*x[ev]+(1+m*z)*x[ev+1])*exp(m*z)+1e-15;
contr[4]:=contr[4]+((1-m*z)*x[ev+3]-m*x[ev+2])*exp(-m*z);
      contr[4]:=contr[4]*6.25*m*m*bessj1(12.5*m)-1e-15
      end
      End
      End
      Else
      Begin
      If z<>hh1 then
      Begin
      ev:=4*n2-3;
      if r<>0 then
      begin
contr[2]:=(m*x[ev]+(2*mm[n2]+m*z)*x[ev+1])*exp(m*z)-1e-15;
contr[2]:=contr[2]+(m*x[ev+2]+(m*z-2*mm[n2])*x[ev+3])*exp(-m*z);
      contr[3]:=-(m*x[ev]+(1+m*z)*x[ev+1])*exp(m*z)+1e-15;
contr[3]:=b1*(contr[3]+(m*x[ev+2]-(1-m*z)*x[ev+3])*exp(-m*z))/(m*r);
      contr[4]:=(m*x[ev]+(1+m*z)*x[ev+1])*exp(m*z)+1e-15;
contr[4]:=b1*(contr[4]+((1-m*z)*x[ev+3]-m*x[ev+2])*exp(-m*z))/(m*r);
      end
      else
      begin
      contr[2]:=0;
contr[3]:=-(m*x[ev]+(1+m*z)*x[ev+1])*exp(m*z)+m*x[ev+2]*exp(-m*z)
+1e-15;
contr[3]:=6.25*m*m*bessj1(12.5*m)*(contr[3]-(1-m*z)*x[ev+3]*exp(-
m*z));
contr[4]:=(m*x[ev]+(1+m*z)*x[ev+1])*exp(m*z)+(1-m*z)*x[ev+3]*exp(

```

```

-m*z)+1e-15;
contr[4]:=6.25*m*m*bessj1(12.5*m)*(contr[4]-m*x[ev+2]*exp(-m*z));
    end
    End
    Else if n2>1 then
    Begin
        ev:=4*n2-7;
        if r<>0 then
            begin

contr[2]:=(m*x[ev]+(2*mm[n2-1]+m*z)*x[ev+1])*exp(m*z)+m*exp(-m*z)
*x[ev+2]-1e-15;

contr[2]:=b1*(contr[2]+(m*z-2*mm[n2-1])*exp(-m*z)*x[ev+3]);

contr[3]:=-b1*(m*x[ev]+(1+m*z)*x[ev+1])*exp(m*z)/(m*r)+1e-15;

contr[3]:=contr[3]-b1*((1-m*z)*x[ev+3]-m*x[ev+2])*exp(-m*z)/(m*r)
;

contr[4]:=b1*(m*x[ev]+(1+m*z)*x[ev+1])*exp(m*z)/(m*r)+1e-15;

contr[4]:=contr[4]+b1*((1-m*z)*x[ev+3]-m*x[ev+2])*exp(-m*z)/(m*r)
;

                end
                else
                begin
                    contr[2]:=0;
                    contr[3]:=-(m*x[ev]+(1+m*z)*x[ev+1])*exp(m*z)+1e-15;

contr[3]:=contr[3]-((1-m*z)*x[ev+3]-m*x[ev+2])*exp(-m*z);
                    contr[3]:=contr[3]*6.25*m*m*bessj1(12.5*m);
                    contr[4]:=(m*x[ev]+(1+m*z)*x[ev+1])*exp(m*z)+1e-15;

contr[4]:=contr[4]+((1-m*z)*x[ev+3]-m*x[ev+2])*exp(-m*z);
                    contr[4]:=contr[4]*6.25*m*m*bessj1(12.5*m)
                    end
                End;
            End;
            case interf of
                1: p1;
                2: p2
            End;
            {case interf}
        End;
        {Fin case}
    End;

End;

Procedure romberg;
Var
    t1,t2,t13,t13sup,t23,t23sup,t14,t14sup,

```

```

t24,t24sup:array[0..10,0..10] of real;
p1,p2,l1,k1:byte;
x1,hp:real;
bs:boolean;
n1,ki:int;
Begin
for ki:=0 to 5 do
begin
for p2:=0 to 5 do
begin
t23sup[ki,p2]:=0;t24sup[ki,p2]:=0;
t13sup[ki,p2]:=0;t14sup[ki,p2]:=0
end
end;
case incontr of
1:Begin
n1:=1;bs:=false;
if b<1 then
begin
t1[0,0]:=0;t13[0,0]:=0;t14[0,0]:=0;
t13sup[0,0]:=0;t14sup[0,0]:=0;
end
else begin
calculz(b);
t1[0,0]:=(b-a)*contr[1]/2;
t13[0,0]:=(b-a)*contr[3]/2;
t14[0,0]:=(b-a)*contr[4]/2;
case interf of
1:begin
if operation='dimt'then
begin
case itf1 of
12:begin
if kk=1 then
begin
t13sup[0,0]:=(b-a)*contrsup[3]/2;
t14sup[0,0]:=(b-a)*contrsup[4]/2
end
end;
23:begin
if kk=2 then
begin
t13sup[0,0]:=(b-a)*contrsup[3]/2;
t14sup[0,0]:=(b-a)*contrsup[4]/2
end
end;
end
(case itf1)
end
else begin
case itf1 of
12:begin
if (zi[kk]=h[1]) then

```

```

begin
    t13sup[0,0]:=(b-a)*contrsup[3]/2;
    t14sup[0,0]:=(b-a)*contrsup[4]/2
end
end;
23:begin
    if (zi[kk]=h[1]+h[2]) then
        begin
            t13sup[0,0]:=(b-a)*contrsup[3]/2;
            t14sup[0,0]:=(b-a)*contrsup[4]/2
        end
    end
end
end (case itf1)
end;
2:begin
    if operation='dimt'then
        begin
            if (kk=1)or(kk=2) then
                begin
                    t13sup[0,0]:=(b-a)*contrsup[3]/2;
                    t14sup[0,0]:=(b-a)*contrsup[4]/2
                end
            end
        else begin
            if (zi[kk]=h[1])or(zi[kk]=h[1]+h[2]) then
                begin
                    t13sup[0,0]:=(b-a)*contrsup[3]/2;
                    t14sup[0,0]:=(b-a)*contrsup[4]/2
                end
            end
        end
    end
end (case interf)
end;
repeat
    l1:=trunc(exp(n1*ln(2)));
    hp:=(b-a)/l1;x1:=a;
    if b<l then
        begin
            t1[0,n1]:=0;t13[0,n1]:=0;t14[0,n1]:=0;
            t13sup[0,0]:=0;t14sup[0,0]:=0
        end
    else begin
        calculz(b);
        t1[0,n1]:=(b-a)*contr[1]/2;
        t13[0,n1]:=(b-a)*contr[3]/2;
        t14[0,n1]:=(b-a)*contr[4]/2;
        case interf of
            1:begin
                if operation='dimt' then
                    begin

```

```

case itf1 of
  12:begin
    if kk=1 then
      begin
        t13sup[0,n1]:=(b-a)*contrsup[3]/2;
        t14sup[0,n1]:=(b-a)*contrsup[4]/2
      end
    end;
  23:begin
    if kk=2 then
      begin
        t13sup[0,n1]:=(b-a)*contrsup[3]/2;
        t14sup[0,n1]:=(b-a)*contrsup[4]/2
      end
    end
end;          (case itf1)
end
else begin
case itf1 of
  12:begin
    if (zi[kk]=h[1]) then
      begin
        t13sup[0,n1]:=(b-a)*contrsup[3]/2;
        t14sup[0,n1]:=(b-a)*contrsup[4]/2
      end
    end;
  23:begin
    if (zi[kk]=h[1]+h[2]) then
      begin
        t13sup[0,n1]:=(b-a)*contrsup[3]/2;
        t14sup[0,n1]:=(b-a)*contrsup[4]/2
      end
    end
end          (case itf1)
end;
end;
2:begin
If operation='dimt'then
Begin
if (kk=1)or(kk=2) then
begin
t13sup[0,n1]:=(b-a)*contrsup[3]/2;
t14sup[0,n1]:=(b-a)*contrsup[4]/2
end
End
Else begin
if (zi[kk]=h[1])or(zi[kk]=h[1]+h[2]) then
begin
t13sup[0,n1]:=(b-a)*contrsup[3]/2;
t14sup[0,n1]:=(b-a)*contrsup[4]/2
end
end
end

```



```

        end
    end      {interf}
end;
for p2:=1 to l1-1 do
begin
x1:=x1+hp;
calculz(x1);
t1[0,n1]:=t1[0,n1]+contr[1];
t13[0,n1]:=t13[0,n1]+contr[3];
t14[0,n1]:=t14[0,n1]+contr[4];
it:=it+1;gotoxy(42,12);
write(it);gotoxy(48,12);
case interf of
1:begin
if operation='dimt'. then
begin
case itf1 of
12:begin
if kk=1 then
begin
t13sup[0,n1]:=t13sup[0,n1]+contrsup[3];
t14sup[0,n1]:=t14sup[0,n1]+contrsup[4]
end
end;
23:begin
if kk=2 then
begin
t13sup[0,n1]:=t13sup[0,n1]+contrsup[3];
t14sup[0,n1]:=t14sup[0,n1]+contrsup[4]
end
end
end;          {case itf1}
end
else begin
case itf1 of
12:begin
if zi[kk]=h[1] then
begin
t13sup[0,n1]:=t13sup[0,n1]+contrsup[3];
t14sup[0,n1]:=t14sup[0,n1]+contrsup[4]
end
end;
23:begin
if (zi[kk]=h[2]+h[1]) then
begin
t13sup[0,n1]:=t13sup[0,n1]+contrsup[3];
t14sup[0,n1]:=t14sup[0,n1]+contrsup[4]
end
end
end;          {case itf1}
end
end;
end;
end;

```

```

2:begin
  if operation='dimt' then
    begin
      if (kk=1)or(kk=2) then
        begin
          t13sup[0,n1]:=t13sup[0,n1]+contrsup[3];
          t14sup[0,n1]:=t14sup[0,n1]+contrsup[4]
        end
      end
    else begin
      if (zi[kk]=h[1])or(zi[kk]=h[1]+h[2]) then
        begin
          t13sup[0,n1]:=t13sup[0,n1]+contrsup[3];
          t14sup[0,n1]:=t14sup[0,n1]+contrsup[4]
        end
      end
    end
  end
  end      {interf}
end;      {boucle}
t1[0,n1]:=t1[0,n1]*hp;
t13[0,n1]:=t13[0,n1]*hp;
t14[0,n1]:=t14[0,n1]*hp;
  t13sup[0,n1]:=hp*t13sup[0,n1];
  t14sup[0,n1]:=hp*t14sup[0,n1];
for p1:=1 to n1 do
  begin
    k1:=n1-p1;
    l1:=trunc(exp(p1*ln(4)));
    t1[p1,k1]:=(l1*t1[p1-1,k1+1]-t1[p1-1,k1])/(l1-1);
    t13[p1,k1]:=(l1*t13[p1-1,k1+1]-t13[p1-1,k1])/(l1-1);
    t14[p1,k1]:=(l1*t14[p1-1,k1+1]-t14[p1-1,k1])/(l1-1);
    t13sup[p1,k1]:=(l1/(l1-1))*t13sup[p1-1,k1+1];
    t13sup[p1,k1]:=t13sup[p1,k1]-t13sup[p1-1,k1]/(l1-1);
    t14sup[p1,k1]:=(l1/(l1-1))*t14sup[p1-1,k1+1];
    t14sup[p1,k1]:=t14sup[p1,k1]-t14sup[p1-1,k1]/(l1-1);
  end;
  if abs(t13[n1,0]-t13[n1-1,0])<1e-5 then bs:=true
  else if n1<3 then n1:=n1+1
        else bs:=true;
until bs;
f1[kk]:=f1[kk]+t1[n1,0];
f13[kk]:=f13[kk]+t13[n1,0];
f14[kk]:=f14[kk]+t14[n1,0];
if operation='dimt' then
  begin
    if (kk=1)or(kk=2) then
      begin
        fsup13[kk]:=fsup13[kk]+t13sup[n1,0];
        fsup14[kk]:=fsup14[kk]+t14sup[n1,0];
      end
    end
  end
else

```

```

        if (zi[kk]=h[1])or(zi[kk]=h[1]+h[2]) then
            begin
                fsup13[kk]:=fsup13[kk]+t13sup[n1,0];
                fsup14[kk]:=fsup14[kk]+t14sup[n1,0];
            end
    End;

2:Begin
    n1:=1;bs:=false;
    if b<1 then
        begin
            t2[0,0]:=0;t23[0,0]:=0;t24[0,0]:=0;
            t23sup[0,0]:=0;t24sup[0,0]:=0;
        end
    else begin
        calculz(b);
        t2[0,0]:=(b-a)*contr[2]/2;
        t23[0,0]:=(b-a)*contr[3]/2;
        t24[0,0]:=(b-a)*contr[4]/2;
        case interf of
            1:begin
                if operation='dimt'then
                    begin
                        case itf1 of
                            12:begin
                                if kk=1 then
                                    begin
                                        t23sup[0,0]:=(b-a)*contrsup[3]/2;
                                        t24sup[0,0]:=(b-a)*contrsup[4]/2
                                    end
                                end;
                            23:begin
                                if kk=2 then
                                    begin
                                        t23sup[0,0]:=(b-a)*contrsup[3]/2;
                                        t24sup[0,0]:=(b-a)*contrsup[4]/2
                                    end
                                end;
                            end
                        end
                    end
                (case itf1)
            end
        else begin
            case itf1 of
                12:begin
                    if (zi[kk]=h[1]) then
                        begin
                            t23sup[0,0]:=(b-a)*contrsup[3]/2;
                            t24sup[0,0]:=(b-a)*contrsup[4]/2
                        end
                    end;
                23:begin
                    if (zi[kk]=h[1]+h[2]) then
                        begin

```

```

                                t23sup[0,0]:=(b-a)*contrsup[3]/2;
                                t24sup[0,0]:=(b-a)*contrsup[4]/2
                                end
                                end
                                end          (case itf1)
                                end
end;
2:begin
  if operation='dimt'then
    begin
      if (kk=1)or(kk=2) then
        begin
          t23sup[0,0]:=(b-a)*contrsup[3]/2;
          t24sup[0,0]:=(b-a)*contrsup[4]/2
        end
      end
    else begin
      if (zi[kk]=h[1])or(zi[kk]=h[1]+h[2]) then
        begin
          t23sup[0,0]:=(b-a)*contrsup[3]/2;
          t24sup[0,0]:=(b-a)*contrsup[4]/2
        end
      end
    end
  end          (case interf)
end;
repeat
  l1:=trunc(exp(n1*ln(2)));
  hp:=(b-a)/l1;x1:=a;
  if b<l then
    begin
      t2[0,n1]:=0;t23[0,n1]:=0;t24[0,n1]:=0 ;
      t13sup[0,0]:=0;t14sup[0,0]:=0
    end
  else begin
    calculz(b);
    t2[0,n1]:=(b-a)*contr[2]/2;
    t23[0,n1]:=(b-a)*contr[3]/2;
    t24[0,n1]:=(b-a)*contr[4]/2;
    case interf of
      1:begin
        if operation='dimt' then
          begin
            case itf1 of
              12:begin
                if kk=1 then
                  begin
                    t23sup[0,n1]:=(b-a)*contrsup[3]/2;
                    t24sup[0,n1]:=(b-a)*contrsup[4]/2
                  end
                end;
              23:begin

```

```

        if kk=2 then
            begin
                t23sup[0,n1]:=(b-a)*contrsup[3]/2;
                t24sup[0,n1]:=(b-a)*contrsup[4]/2
            end
        end
    end;          (case itf1)
    end
    else begin
    case itf1 of
        12:begin
            if (zi[kk]=h[1]) then
                begin
                    t23sup[0,n1]:=(b-a)*contrsup[3]/2;
                    t24sup[0,n1]:=(b-a)*contrsup[4]/2
                end
            end;
        23:begin
            if (zi[kk]=h[1]+h[2]) then
                begin
                    t23sup[0,n1]:=(b-a)*contrsup[3]/2;
                    t24sup[0,n1]:=(b-a)*contrsup[4]/2
                end
            end
        end
    end          (case itf1)
    end;
2:begin
    If operation='dimt'then
        Begin
            if (kk=1)or(kk=2) then
                begin
                    t23sup[0,n1]:=(b-a)*contrsup[3]/2;
                    t24sup[0,n1]:=(b-a)*contrsup[4]/2
                end
            end
        End
    Else begin
        if (zi[kk]=h[1])or(zi[kk]=h[1]+h[2]) then
            begin
                t23sup[0,n1]:=(b-a)*contrsup[3]/2;
                t24sup[0,n1]:=(b-a)*contrsup[4]/2
            end
        end
    end
end          (interf)
end;
for p2:=1 to l1-1 do
    begin
        x1:=x1+hp;
        calculz(x1);
        t2[0,n1]:=t2[0,n1]+contr[2];
        t23[0,n1]:=t23[0,n1]+contr[3];
    end
end

```

```

t24[0,n1]:=t24[0,n1]+contr[4];
it:=it+1;gotoxy(42,12);
write(it);gotoxy(48,12);
case interf of
  1:begin
    if operation='dimt' then
      begin
        case itf1 of
          12:begin
            if kk=1 then
              begin
                t23sup[0,n1]:=t23sup[0,n1]+contrsup[3];
                t24sup[0,n1]:=t24sup[0,n1]+contrsup[4]
              end
            end;
          23:begin
            if kk=2 then
              begin
                t23sup[0,n1]:=t23sup[0,n1]+contrsup[3];
                t24sup[0,n1]:=t24sup[0,n1]+contrsup[4]
              end
            end
          end;
        end;
      end;
    else begin
      case itf1 of
        12:begin
          if zi[kk]=h[1] then
            begin
              t23sup[0,n1]:=t23sup[0,n1]+contrsup[3];
              t24sup[0,n1]:=t24sup[0,n1]+contrsup[4]
            end
          end;
        23:begin
          if (zi[kk]=h[2]+h[1]) then
            begin
              t23sup[0,n1]:=t23sup[0,n1]+contrsup[3];
              t24sup[0,n1]:=t24sup[0,n1]+contrsup[4]
            end
          end
        end;
      end;
    end;
  2:begin
    if operation='dimt' then
      begin
        if (kk=1)or(kk=2) then
          begin
            t23sup[0,n1]:=t23sup[0,n1]+contrsup[3];
            t24sup[0,n1]:=t24sup[0,n1]+contrsup[4]
          end
        end
      end
    end
  end
end

```

```

        else begin
            if (zi[kk]=h[1]) or (zi[kk]=h[1]+h[2]) then
                begin
                    t23sup[0,n1]:=t23sup[0,n1]+contrsup[3];
                    t24sup[0,n1]:=t24sup[0,n1]+contrsup[4]
                end
            end
        end
    end
end
(interf)
end;
t2[0,n1]:=t2[0,n1]*hp;
t23[0,n1]:=t23[0,n1]*hp;
t24[0,n1]:=t24[0,n1]*hp;
    t23sup[0,n1]:=hp*t23sup[0,n1];
    t24sup[0,n1]:=hp*t24sup[0,n1];
for p1:=1 to n1 do
    begin
        k1:=n1-p1;
        l1:=trunc(exp(p1*ln(4)));
        t2[p1,k1]:=(l1*t2[p1-1,k1+1]-t2[p1-1,k1])/(l1-1);
        t23[p1,k1]:=(l1*t23[p1-1,k1+1]-t23[p1-1,k1])/(l1-1);
        t24[p1,k1]:=(l1*t24[p1-1,k1+1]-t24[p1-1,k1])/(l1-1);
        t23sup[p1,k1]:=(l1/(l1-1))*t23sup[p1-1,k1+1];
        t23sup[p1,k1]:=t23sup[p1,k1]-t23sup[p1-1,k1]/(l1-1);
        t24sup[p1,k1]:=(l1/(l1-1))*t24sup[p1-1,k1+1];
        t24sup[p1,k1]:=t24sup[p1,k1]-t24sup[p1-1,k1]/(l1-1);
    end;
if abs(t23[n1,0]-t23[n1-1,0])<1e-5 then bs:=true
else if n1<3 then n1:=n1+1
    else bs:=true;
until bs;
f2[kk]:=f2[kk]+t2[n1,0]; f23[kk]:=f23[kk]+t23[n1,0];
f24[kk]:=f24[kk]+t24[n1,0];
if operation='dimt' then
    begin
        if (kk=1) or (kk=2) then
            begin
                fsup23[kk]:=fsup23[kk]+t23sup[n1,0];
                fsup24[kk]:=fsup24[kk]+t24sup[n1,0];
            end
        end
    else
        if (zi[kk]=h[1]) or (zi[kk]=h[1]+h[2]) then
            begin
                fsup23[kk]:=fsup23[kk]+t23sup[n1,0];
                fsup24[kk]:=fsup24[kk]+t24sup[n1,0];
            end
        end
    End
End;
(fin case)
f3[kk]:=f13[kk]+f23[kk]; f4[kk]:=f14[kk]+f24[kk];
If operation='dimt' then
    begin

```

```

    if (kk=1)or(kk=2) then
      begin
        fsup3[kk]:=fsup13[kk]+fsup23[kk];
        fsup4[kk]:=fsup14[kk]+fsup24[kk];
      end
    end
  Else begin
    if (zi[kk]=h[1])or(zi[kk]=h[1]+h[2]) then
      begin
        fsup3[kk]:=fsup13[kk]+fsup23[kk];
        fsup4[kk]:=fsup14[kk]+fsup24[kk];
      end
    end;
  End;

Procedure integration;
Var
  i1,i2:byte;
  ki:int;
Begin
  if z<5 then
    begin
      if r<12.5 then l:=125  else l:=3*r
    end
  else
    begin
      if z<15 then if r>12.5 then l:=2*r else l:=25
      else if r>12.5 then l:=r else l:=12.5;
    end;
  nbvaleur:=trunc(1/3.13);
  for ki:=0 to 6 do
    begin
      f13[ki]:=0;f23[ki]:=0;f14[ki]:=0;
      f24[ki]:=0;fsup13[ki]:=0;fsup23[ki]:=0;
      fsup14[ki]:=0;fsup24[ki]:=0;
    end;
  For incontr:=1 to 2 do
    Begin
      i1:=1;i2:=1;a:=0;
      Case incontr of
        1: Begin
          If r<>0 then
            Begin
              while (i1<=nbvaleur)and(i2<=nbvaleur) do
                begin
                  if (rac11[i1]/r)<(rac12[i2]/12.5) then
                    begin
                      b:=rac11[i1]/r;i1:=i1+1;
                      romberg;a:=b
                    end
                  else begin
                    b:=rac12[i2]/12.5;i2:=i2+1;

```



```

                romberg;a:=b
            end
        end;
        if r>12.5 then b:=1/r
        else b:=1/12.5;
            romberg;a:=b
        End
    Else begin
        while i1<=nbvaleur do
            begin
                b:=rac12[i1]/12.5;i1:=i1+1;
                romberg;a:=b
            end;
            b:=1/12.5;
            romberg;a:=b
        end
    End;

2: Begin
    If r<>0 then
        Begin
            while (i1<=nbvaleur)and(i2<=nbvaleur) do
                begin
                    if (rac12[i1]/r)<(rac12[i2]/12.5) then
                        begin
                            b:=rac12[i1]/r;i1:=i1+1;
                            romberg;a:=b
                        end
                    else begin
                            b:=rac12[i2]/12.5;i2:=i2+1;
                            romberg;a:=b
                        end
                    end;
                    if r>12.5 then b:=1/r
                    else b:=1/12.5;
                    romberg;a:=b
                End
            Else begin
                while i1<=nbvaleur do
                    begin
                        b:=rac12[i1]/12.5;i1:=i1+1;
                        romberg;a:=b
                    end;
                    b:=1/12.5;
                    romberg;a:=b
                end
            End
        End
    End;
    End;
End.
    ( Fin case)

```

```

Unit unite4;(Calcul et Dimensionnement)
  Interface
uses crt,unite2,unite3,unite5;

  Procedure type_operat;

      Implementation
  Procedure cadre2(i1,j1,i2,j2:byte);
  var
    i:byte;
  Begin
    gotoxy(j1,i1);write(chr(201));
    for i:=j1+1 to j2-1 do
      Begin
        gotoxy(i,i1);write(chr(205));
        gotoxy(i,i2);write(chr(205));
      End;
    gotoxy(j2,i1);writeln(chr(187));
    gotoxy(j1,i2);write(chr(200));
    for i:=i1+1 to i2-1 do
      begin
        gotoxy(j1,i);write(chr(186));
        gotoxy(j2,i);write(chr(186));
      end;
    gotoxy(j2,i2);writeln(chr(188));
  End;

```

```

Procedure cad_band(i1,j1,i2,j2:shortint);
Var
  i:shortint;
Begin
  textcolor(yellow);textbackground(black);
  for i:=j1 to j2 do
    Begin
      gotoxy(i,i1);write(chr(220));
      gotoxy(i,i2);write(chr(220));
    End;
  for i:=i1+1 to i2 do
    begin
      gotoxy(j1,i);write(chr(219));
      gotoxy(j2,i);write(chr(219));
    end;
  normvideo
End;

```

```

Procedure nbrecouches;

```

```

Begin
  textcolor(0);textbackground(7);
  gotoxy(44,6);write('Indications');
  textcolor(7+blink);textbackground(0);
  gotoxy(36,8);write(' ! ');
  textcolor(yellow);textbackground(0);
  gotoxy(38,8);write(' Ce nombre est nécessaire');
  gotoxy(36,9);write(' pour la suite des entrées. ');
  gotoxy(36,10);write(' Il doit être ≥ 2 pour le');
  gotoxy(36,11);write(' dimensionnement. Mais ≤ 5');
  gotoxy(36,12);write(' dans tous les cas. ');
  normvideo;
End;

```

```

Procedure condinterf;

```

```

Begin
  textcolor(0);textbackground(7);
  gotoxy(54,8);write(' A lire ');
  textcolor(yellow);textbackground(0);
  gotoxy(38,10);write(' Entrer 0 s''il n''y a pas de
  glissement. ');
  gotoxy(37,11);write(' 1 s''il en a à une seule interface. ');
  gotoxy(37,12);write(' 2 s''il y a glissement en deux
  interfaces. ');
  gotoxy(38,14);write(' Cette valeur entrée est inférieure ou');
  gotoxy(37,15);write(' égale à 2. ');normvideo;
End;

```

```

Procedure caracteristiques;

```

```

Begin
  textcolor(red);textbackground(yellow);
  cadre2(4,6,16,35);

```

```

gotoxy(14,4);write('Caractéristiques');
normvideo;
gotoxy(8,6);write('Nombre de couches:');
gotoxy(8,8);write('Glissement:');
gotoxy(8,10);write('Coefficients de Poisson');
gotoxy(8,12);write('Modules d''élasticité (Bar)');
gotoxy(8,14);write('Essieu Standard:');
End;

Procedure calculstruc;
Var
  fca1,fca2,fca3,fca4,fcasup3,fcasup4:array[1..5,1..2] of real;
  wca:array[1..5,1..2] of real;
  voi,bs:boolean;
  ki:int;
  rev:char;
Begin
  if r<12.5 then l:=125
  else l:=3*r;
  lecture;
  for ki:=0 to np do
    begin
      f1[ki]:=0;f2[ki]:=0;f3[ki]:=0;
      f4[ki]:=0;fsup3[ki]:=0;fsup4[ki]:=0
    end;
  clrscr;
  textcolor(yellow+blink);textbackground(red);
  gotoxy(70,1);write(' PATIENCE ');
  normvideo;
  Gotoxy(1,1);write('——ZOVROUTE——');
  cad_band(6,20,15,48);
  textcolor(yellow);textbackground(red);
  gotoxy(24,8);write(' Calcul de contraintes ');
  normvideo;
  gotoxy(22,10);write('Point n°:');
  gotoxy(22,12);write('Nombre d''itérations:');
  If dual='0' then
    Begin
      gotoxy(22,14);write('Roues jumelées. ');
      If r<>18.75 then
        Begin
          for kk:=1 to np do
            begin
              it:=0;
              gotoxy(44,10); write(kk);
              gotoxy(42,12);write(it,' ');
              z:=zi[kk];
              integration;
              fca1[kk,1]:=f1[kk];fca2[kk,1]:=f2[kk];
              fca3[kk,1]:=f3[kk];fca4[kk,1]:=f4[kk];
              fcasup3[kk,1]:=fsup3[kk];fcasup4[kk,1]:=fsup4[kk];
            end;
          end;
        end;
      end;
    end;
  end;

```

```

r:=abs(37.5-r);
for ki:=0 to np do
  begin
    f1[ki]:=0;f2[ki]:=0;f3[ki]:=0;
    f4[ki]:=0;fsup3[ki]:=0;fsup4[ki]:=0
  end;
for kk:=1 to np do
  begin
    it:=0;
    gotoxy(44,10); write(kk);
    gotoxy(42,12);write(it, ' ');
    z:=zi[kk];
    integration;
    fca1[kk,2]:=f1[kk];fca2[kk,2]:=f2[kk];
    fca3[kk,2]:=f3[kk];fca4[kk,2]:=f4[kk];
    fcasup3[kk,2]:=fsup3[kk];fcasup4[kk,2]:=fsup4[kk];
  end;
End
Else
begin
  for ki:=0 to np do
    begin
      f1[ki]:=0;f2[ki]:=0;f3[ki]:=0;
      f4[ki]:=0;fsup3[ki]:=0;fsup4[ki]:=0
    end;
  for kk:=1 to np do
    begin
      it:=0;
      gotoxy(44,10); write(kk);
      gotoxy(42,12);write(it, ' ');
      z:=zi[kk];
      integration;
      fca1[kk,1]:=f1[kk];fca2[kk,1]:=f2[kk];
      fca3[kk,1]:=f3[kk];fca4[kk,1]:=f4[kk];
      fcasup3[kk,1]:=fsup3[kk];fcasup4[kk,1]:=fsup4[kk];
      fca1[kk,2]:=fca1[kk,1];fca2[kk,2]:=fca2[kk,1];
      fca3[kk,2]:=fca3[kk,1];fca4[kk,2]:=fca4[kk,1];
      fcasup3[kk,2]:=fcasup3[kk,1];fcasup4[kk,2]:=fcasup4[kk,1];
    end
  end;
  for kk:=1 to np do
    begin
      f1[kk]:=fca1[kk,1]+fca1[kk,2];f2[kk]:=fca2[kk,1]+fca2[kk,2];
      f3[kk]:=fca3[kk,1]+fca3[kk,2];f4[kk]:=fca4[kk,1]+fca4[kk,2];
      fsup3[kk]:=fcasup3[kk,1]+fcasup3[kk,2];
      fsup4[kk]:=fcasup4[kk,1]+fcasup4[kk,2];
    end
  End
Else

```

```

Begin
  for ki:=0 to np do
    begin
      f1[ki]:=0;f2[ki]:=0;f3[ki]:=0;
      f4[ki]:=0;fsup3[ki]:=0;fsup4[ki]:=0
    end;
    gotoxy(22,14);write('Roue simple. ');
    for kk:=1 to np do
      begin
        it:=0;
        gotoxy(44,10); write(kk);
        gotoxy(42,12);write(it,' ');
        z:=zi[kk];
        integration;
      end;
    End;
  gotoxy(1,1);clreol;
  textcolor(7+blink);textbackground(0);
  gotoxy(20,20);write('Operation terminée ');
  normvideo;
  gotoxy(15,22);write('Appuyer une touche');
  rev:=readkey;
  textcolor(white);textbackground(blue);
  clrscr;affichage;
  gotoxy(1,24);
  for k:=1 to 78 do write('-');
  normvideo;
  repeat
    textcolor(white);textbackground(red);
    gotoxy(1,25); clreol;
    gotoxy(15,25);
    write(' F2-sauvegarde          F4-impression          F10-Menu');
    gotoxy(75,25);
    c:=readcar(bs);
    normvideo;
    if c=#60 then nomfich;
    if c=#62 then
      begin
        gotoxy(1,25);clreol;
        gotoxy(15,24); textcolor(white);textbackground(red);
        write('Vérifiez l'imprimante et appuyez une touche');
        rev:=readkey;
        normvideo;
        impression;
      end;
    if c=#68 then
      begin
        clrscr;
        type_operat
      end;
    until (c=#68) and bs;
  End;          {Fin calculstruc}

```

```

Procedure dimenstruc;
Var
  fdi1,fdi2,fdi3,fdi4,fdisup3,fdisup4:array[0..5,1..2,1..2] of
real;
  a2,hor,nft,ne:array[1..2] of real;
  ao,nsol:array[0..5,1..2] of real;
  cl,ep,qot,norn:real;
  id,kp,kc,ki:int;
  bs:boolean;
  rev:char;
Begin
  clrscr;
  gotoxy(15,22);writeln('Utiliser ',chr(26),' ',chr(27),' pour le
choix');
  gotoxy(15,23);
  writeln(' et ',chr(24),chr(25),' pour augmenter ou diminuer');
  gotoxy(15,24);writeln(' Entrer enfin de toutes les
modifications');
  selecthauteur;
  h[nc]:=0;for id:=1 to nc-1 do h[nc]:=h[nc]+h[id];
  if h[nc]>70 then
    Begin
      write(#7);gotoxy(10,3);clrscr;
      textcolor(7+blink);textbackground(red);
      write(' Superstructure trop grande ');
      normvideo;write('Fin des opérations');
      delay(5000);type_operat;
    End
  Else Begin
    textcolor(yellow+blink);textbackground(red);
    gotoxy(70,1);write(' PATIENCE ');
    normvideo;
    Gotoxy(1,1);write('—ZOVROUTE—');
    cad_band(6,20,15,48);
    textcolor(yellow);textbackground(red);
    gotoxy(27,8);write(' Dimensionnement ');
    normvideo;
    gotoxy(22,12);write('Nombre d''itérations:');
    gotoxy(42,12);write(' ');
    mdl[1]:=el[1];mdl[2]:=mo;
    For ki:=1 to 2 do
      Begin
        it:=0;
        for id:=0 to nc do
          begin
            f1[id]:=0;f2[id]:=0;f3[id]:=0;
            f4[id]:=0;fsup3[id]:=0;fsup4[id]:=0
          end;
        if ki=2 then el[1]:=mo;
        gotoxy(22,10);write('Période n°: ',ki);
        gotoxy(42,12);write(it,' ');
        If dual='0' then

```

```

Begin
  gotoxy(22,14);write('Roues jumelées. ');
  r:=0;z:=0;
  for kk:=0 to nc-1 do
    begin
      if kk=0 then z:=0
      else z:=z+h[kk];
      integration;
      fdi1[kk,ki,1]:=f1[kk];fdi2[kk,ki,1]:=f2[kk];
      fdi3[kk,ki,1]:=f3[kk];fdi4[kk,ki,1]:=f4[kk];
      fdisup3[kk,ki,1]:=fsup3[kk];
      fdisup4[kk,ki,1]:=fsup4[kk];
    end;
  r:=37.5;z:=0;
  for id:=0 to nc do
    begin
      f1[id]:=0;f2[id]:=0;f3[id]:=0;
      f4[id]:=0;fsup3[id]:=0;fsup4[id]:=0
    end;
  for kk:=0 to nc-1 do
    begin
      if kk=0 then z:=0 else z:=z+h[kk];
      integration;
      fdi1[kk,ki,2]:=f1[kk];fdi2[kk,ki,2]:=f2[kk];
      fdi3[kk,ki,2]:=f3[kk];fdi4[kk,ki,2]:=f4[kk];
      fdisup3[kk,ki,2]:=fsup3[kk];
      fdisup4[kk,ki,2]:=fsup4[kk];
    end;
  for kk:=0 to nc-1 do
    begin
      fd1[kk,ki]:=fdi1[kk,ki,1]+fdi1[kk,ki,2];
      fd2[kk,ki]:=fdi2[kk,ki,1]+fdi2[kk,ki,2];
      fd3[kk,ki]:=fdi3[kk,ki,1]+fdi3[kk,ki,2];
      fd4[kk,ki]:=fdi4[kk,ki,1]+fdi4[kk,ki,2];
      fdsup3[kk,ki]:=fdisup3[kk,ki,1]+fdisup3[kk,ki,2];
      fdsup4[kk,ki]:=fdisup4[kk,ki,1]+fdisup4[kk,ki,2];
    end
  End
Else
Begin
  gotoxy(22,14);write('Roue simple. ');
  r:=0;
  gotoxy(42,12);write(it, ' ');
  for kk:=0 to nc-1 do
    begin
      if kk=0 then z:=0 else z:=z+h[kk];
      integration;
      fd1[kk,ki]:=f1[kk];fd2[kk,ki]:=f2[kk];
      fd3[kk,ki]:=f3[kk];fd4[kk,ki]:=f4[kk];
      fdsup3[kk,ki]:=fsup3[kk];fdsup4[kk,ki]:=fsup4[kk];
    end
  End;

```



```

End;                                (boucle ki)
For kp:=1 to 2 do
  Begin
    If ct='0' then
      Begin
        for kc:=nc downto 3 do
          begin
            c1:=(0.03*el[kc]/(abs(qo*fd1[kc-1,kp]))-10)*ln(10)/7;
(100CBR)
            if c1<=60 then nsol[kc,kp]:=exp(c1) else
nsol[kc,kp]:=1e+20;
            end;
            i f f d 3 [ 2 , k p ] > 0 t h e n
nft[kp]:=exp(20*(1-qo*fd3[2,kp]/cr)*ln(10))
            else nft[kp]:=1e+20;
          End
        Else
          Begin
            for kc:=nc downto 2 do
              begin
c1:=(0.03*el[kc]/(abs(qo*fd1[kc-1,kp]))-10)*ln(10)/7;
              if c1<=60 then nsol[kc,kp]:=exp(c1) else
nsol[kc,kp]:=1e+20;
              end;
            End;
            ep:=qo*abs(fd3[1,kp]-mm[1]*(fd4[1,kp]+fd1[1,kp]))/md1[kp];
            if ep>=5e-9 then ne[kp]:=exp(ln(0.0016/ep)/0.21) else
ne[kp]:=1e+20;
            for id:=0 to 1 do
              ao[id,kp]:=qo*(abs(fd1[id,kp])-abs(fd3[id,kp]));
            End;
            (kp)
          For kp:=1 to 2 do
            Begin
              If (ao[0,kp]<0)and(ao[1,kp]>0) then
                begin
                  hor[kp]:=ao[1,kp]*h[1]/(abs(ao[0,kp])+ao[1,kp]);
                  a2[kp]:=ao[1,kp];
                end
              Else if ao[0,kp]>0 then
                begin
                  hor[kp]:=h[1];
                  a2[kp]:=ao[0,kp]+ao[1,kp];
                end
              else hor[kp]:=0;
            End;
            got:=0;
          For kp:=1 to 2 do
            Begin
              got:=got+hor[kp]*a2[kp]*exp(0.25*ln(a1[kp]))/md1[kp];
            End;
          If got>0 then norn:=0.0018525*vm*exp(-4*ln(got))

```

```

Else norn:=1e+20;
cyc[0]:=norn;
If ct='0' then
  Begin
    For id:=3 to nc do
      cyc[id]:=nsol[id,1]*(nsol[id,2]/(a1[1]*nsol[id,2]+
      a1[2]*nsol[id,1]));
      cyc[2]:=nft[1]*(nft[2]/(a1[1]*nft[2]+a1[2]*nft[1]));
    End
  Else
  Begin
    for id:=2 to nc do
cyc[id]:=nsol[id,1]*(nsol[id,2]/(a1[1]*nsol[id,2]+a1[2]*nsol[id,1
]));
  End;
  cyc[1]:=ne[1]*(ne[2]/(a1[1]*ne[2]+a1[2]*ne[1]));
  gotoxy(1,1);clreol;
  textcolor(7+blink);
  gotoxy(20,20);write('Operation terminée ');
  normvideo;
  gotoxy(15,22);write('Appuyez une touche');
  c:=readcar(bs);
  textcolor(white);textbackground(blue);
  clrscr;affichage;normvideo;
  prendre(15,20,2,40);
  gotoxy(22,16);write('Résultats satisfaisants O/N ? ');
  gotoxy(53,16);rev:=upcase(readkey); write(c);delay(100);
  restitue(15,20,2,40);
  if rev='0' then
  begin
    gotoxy(1,24);
    for k:=1 to 78 do write('-');
    repeat
      textcolor(white);textbackground(red);
      gotoxy(1,25);clreol;
      gotoxy(15,25);
      write(' F2-sauvegarde          F4-impression          F10-Menu');
      gotoxy(75,25); normvideo;
      c:=readcar(bs);
      if bs then
      Case c of
        #60 :nomfich;
        #62 :
          begin
            gotoxy(1,25);clreol;
            gotoxy(15,24);textcolor(white);textbackground(red);
            write('Vérifiez l'imprimante et appuyez une touche');
            rev:=readkey; normvideo;
            impression;
          end;
        #68 : begin

```

```

                clrscr;
                type_operat;
            end;
        End;
    until (c=#68) and bs
end
else begin
    clrscr;
    dimenstruc;
end
    End; {fin de limitation sur  $\Sigma_i$ }
End;

Procedure select;
var
    c,rev,tou,c2:char;
    a,choi,i,ki:int;
    bs,bi:boolean;
    v:real;
PROCEDURE lire(x,y:shortint);
type
    dom=-5..20;
var
    ii:dom;
    mot:string;
    carac:char;
    bs1,ok:boolean;
    erreur:integer;
Begin
    ok:=false;mot:='';ta:=0;
    Gotoxy(x,y);c2:=readcar(bs1);
    If (c2 in ['0'..'9','+']) then
        begin
            mot:=c2;
            textcolor(0); textbackground(7);
            write(c2);normvideo;
            Repeat
                ii:=0;textcolor(0); textbackground(7);
                Repeat
                    carac:=readcar(bs1);
                    if not ((carac in [#72,#80]) or bs1) then
                        begin
                            gotoxy(x+ii+1,y);write(carac);
                        end;
                    if carac=#8 then
                        begin
                            if ii>=0 then
                                begin
                                    ii:=ii-1;
                                    gotoxy(x+ii+1,y);write(' ');
                                    Delete(mot,ii+2,2);
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end

```

```

        else write(#7)
        end
        else ii:=ii+1;
        gotoxy(x+ii+1,y);
        if not ((carac in [#8,#13]) or bs1) then mot:=mot+carac;
Until carac in [#13,#72,#80,#27];
normvideo;
if carac=#27 then
begin repeat type_operat;until c1='0';halt;end;
val(mot,ta,erreur);
if erreur<>0 then
begin
gotoxy(x,y);write(' ':length(mot));
write(#7);gotoxy(x,y);
end
else ok:=true;
mot:='';normvideo;
Until ok;
end;
if c2=#27 then begin repeat type_operat until
c1='0';clrscr;halt end
End;
Begin
clrscr;
textcolor(0);textbackground(7);
gotoxy(21,1);
if operation='dimt' then write(' Dimensionnement ')
else write(' Calcul de contraintes ');
normvideo;gotoxy(1,2);
bi:=true;
textcolor(yellow);for k:=1 to 78 do write('=');normvideo;
gotoxy(15,21);write('Utiliser ',chr(24),' ',chr(25),' pour le
choix.');
```

```

gotoxy(15,22);write('Et valider avec la touche "Entrée");
gotoxy(10,23);write(' F8-Sortie du cadre Esc: Ecran
précéd');
```

```

caracteristiques;tou:='b';
a:=6;choi:=a;
repeat
if bs then
Case tou of
#72: Begin
If a>6 then a:=a-2
else if a=6 then a:=14;
choi:=a;
End;
#80: Begin
If a<14 then a:=a+2
else if a=14 then a:=6;
choi:=a;
End;
End; {Case)

```

Case choi of

```
6: begin
  textcolor(yellow);
  prendre(6,35,7,29);
  normvideo;
  nbrecouches;
  repeat
    textcolor(0);textbackground(7);
    gotoxy(29,6);write(' ');
    lire(30,6);
    if not (c2 in [#72,#80]) then nc:=trunc(ta);
    if not (nc in entier) then
      begin
        write(#7);
        bi:=false;
      end
    else If operation='dimt' then
      Begin
        if nc<2 then
          begin
            write(#7);
            bi:=false;
          end
        else bi:=true;
      End
    Else bi:=true;
  until bi;
  textcolor(7);textbackground(0);
  gotoxy(29,6);write(' ');
  gotoxy(30,6);write(nc);
  restitue(6,35,7,29);normvideo;
end;
8: begin
  textcolor(yellow);
  prendre(8,35,8,43);
  normvideo;
  condinterf;
  textcolor(0);textbackground(7);
  repeat
    gotoxy(29,8);write(' ');
    lire(30,8);
    if not (c2 in [#72,#80]) then interf:=trunc(ta);
    if not (interf in [0,1,2]) then write(#7);
  until (interf in [0,1,2]);
  textcolor(7);textbackground(0);
  gotoxy(29,8);write(' ');
  gotoxy(30,8);write(interf);
  restitue(8,35,8,43);
  normvideo;
  if interf=1 then
    begin
      prendre(8,35,2,24);
```

```

        gotoxy(36,9);write(' Interface 1/2 o/n ? ');
        rev:=upcase(readkey);
        write(rev); delay(200);
        if rev='O' then itf1:=12 else itf1:=23;
        restitue(8,35,2,24);
    end;
end;
10:begin
    textcolor(0);textbackground(7);
    gotoxy(8,10);write('Coefficients de Poisson');
    normvideo;
    prendre(10,35,2*nc,13);
    for k:=1 to nc do
        begin
            gotoxy(37,9+2*k);write('Mu',k,'=');
        end;
    for k:=1 to nc do
        begin
            repeat
                textcolor(0);textbackground(7);
                lire(42,9+2*k);
                if not (c2 in [#72,#80]) then mm[k]:=ta;
                if (mm[k]<0.2)or(mm[k]>0.5) then
                    begin
                        normvideo;write(#7);gotoxy(10,20);
                        write('Toujours ≤0.5 et ≥0.2');
                        delay(1000);gotoxy(10,20);clreol
                    end;
                until (mm[k]>=0.2)and(mm[k]<=0.5);
                normvideo;gotoxy(42,9+2*k);write(' ');
                gotoxy(42,9+2*k);write(mm[k]:4:2);
            end;delay(500);
            restitue(10,35,2*nc,13);
            gotoxy(20,10);write(chr(219));gotoxy(20,10);
            repeat until keypressed;
            gotoxy(8,10);write('Coefficients de Poisson');
        end;
    end;
12:begin
    textcolor(0);textbackground(7);
    gotoxy(8,12);write('Modules d''élasticité (Bar)');
    normvideo;
    prendre(12,35,2*nc,16);
    if operation='dimt'then
        begin
            textcolor(0+blink);textbackground(7);
            gotoxy(37,12);write('Période froide');
            normvideo;
        end;
    for k:=1 to nc do
        begin
            gotoxy(37,11+2*k);write('E',k,'=');
        end;
    end;

```

```

for k:=1 to nc do
  begin
    repeat
      textcolor(0);textbackground(7);
      lire(41,11+2*k);
      if not (c2 in [#72,#80]) then el[k]:=ta;
      if el[k] <=0 then
        begin
          normvideo;write(#7);gotoxy(10,20);
          write('Valeur incompatible');
          delay(600);gotoxy(10,20);
          clreol;
        end;
      until el[k]>0;
      normvideo;gotoxy(41,11+2*k);write(' ');
      gotoxy(41,11+2*k);write(el[k]:8:0)
    end;delay(500);
    restitue(12,35,2*nc,16);
    gotoxy(15,12);write(chr(219));gotoxy(15,12);
    repeat until keypressed;
    gotoxy(8,12);write('Modules d''élasticité (Bar)');
  end;
14:begin
  textcolor(0);textbackground(7);
  gotoxy(8,14);write('Essieu standard ');
  normvideo;
  prendre(14,35,2,10);
  gotoxy(37,17);write('=charge en tonnes');
  repeat
    textcolor(0);textbackground(7);
    gotoxy(37,15);write(' ');
    lire(37,15);
    if not (c2 in [#72,#80]) then qo:=ta;
    if qo <=0 then
      begin
        normvideo;write(#7);gotoxy(10,20);
        write('Valeur incompatible');
        delay(600);gotoxy(10,20);clreol;
      end;
    until qo>0;
    normvideo;
    gotoxy(37,17);clreol;
    restitue(14,35,2,10);
    prendre(14,35,2,24);
    gotoxy(37,15);write('Roues jumelées o/n? ');
    gotoxy(56,15);dual:=upcase(readkey);
    gotoxy(57,15);write(dual);delay(200);
    restitue(14,35,2,24);
    gotoxy(14,14);write(chr(219));gotoxy(14,14);
    if dual='O' then qo:=0.5093*qo {100*(10*qo)/(4A)}
    else qo:=1.0186*qo; {100*(10*qo)/(2A)}
  repeat until keypressed;

```

```

        gotoxy(8,14);write('Essieu standard:');
    end;
End;          {Case}
c:=readcar(bs);tou:=c;
until c in [#66,#27];
if c=#27 then begin repeat type_operat;until
cl='0';clrscr;halt;end;
if operation='dimt' then
begin
for i:=1 to 3 do begin gotoxy(10,20+i);clreol;end;
gotoxy(15,22);textcolor(0);textbackground(7);
write('Entrer le module (bar) de la première couche ');
gotoxy(15,23);write('(couche bitumineuse) pour la période
chaude:');
normvideo;gotoxy(60,23);clreol;
repeat lire(60,23);if not (c2 in [#72,#80]) then mo:=ta;
if mo<=0 then write(#7); until mo>0;
gotoxy(15,22);clreol;gotoxy(15,23);clreol;
textcolor(0);textbackground(7);
gotoxy(10,18);write('Vitesse moyenne du trafic (km/h): ');
normvideo;
repeat lire(44,18);if not (c2 in [#72,#80]) then vm:=ta;
if vm<=0 then write(#7); until vm>0;vm:=vm/3.6;
gotoxy(10,18);clreol;
textcolor(0);textbackground(7);
write('Couche de base traitée au ciment o/n? ');
normvideo;
ct:=upcase(readkey);gotoxy(48,18);
write(ct);delay(100);
if ct='0' then
begin
gotoxy(10,18);clreol;
textcolor(0);textbackground(7);
write('Résistance à la rupture (bar) en flexion ');
normvideo;textcolor(0);textbackground(7);
gotoxy(10,19);write('du matériau traité au ciment ? ');
normvideo;
repeat lire(42,19);if not (c2 in [#72,#80]) then cr:=ta;
if cr<=0 then write(#7);until cr>0;
end;
gotoxy(10,18);clreol;gotoxy(10,19);clreol;
for i:=1 to 2 do
begin
gotoxy(10,19);write('Pourcentage ≤ 1');
textcolor(0);textbackground(7);
gotoxy(10,18);
if i=1 then write('Pourcentage du trafic en période
froide: ')
else write('Pourcentage du trafic en période chaude: ');
normvideo;gotoxy(52,18);clreol;
repeat lire(52,18);if not (c2 in [#72,#80]) then
a1[i]:=ta;

```



```

        if not ((a1[i]<=1) and (a1[i]>=0)) then write(#7);
        until (a1[i]>=0)and(a1[i]<=1);
        gotoxy(10,19);clreol; gotoxy(10,18);clreol;
    end;
gotoxy(1,24);
for k:=1 to 78 do write('-');
gotoxy(10,25);
write(' Esc-écran préc          F7-exéc          F10-Menu');
gotoxy(75,25);
repeat
    c:=readcar(bs);
until c in [#27,#60,#65,#68];
if c=#27 then select;
if c=#65 then
    begin
        textcolor(white);textbackground(red);
        prendre(18,20,3,45);
        gotoxy(22,19);write(' Inserez la disquette programme
dans');
        gotoxy(22,20);write(' l''unité A et appuyez une
touche');
        rev:=readkey;
        restitue(18,20,3,45);
        normvideo;
        clrscr;
        l:=125;
        lecture;
        dimenstruc;
    end;
if c=#68 then
    begin
        clrscr;
        type_operat;
    end;
end
else
begin
for i:=1 to 3 do begin gotoxy(10,20+i);clreol;end;
prendre(20,20,2,36);
gotoxy(21,21);write('Epaisseur (cm) de la couche ');
for i:=1 to nc-1 do
    begin
        repeat
            gotoxy(49,21);write(i,':      ');
            textcolor(0);textbackground(7);
            lire(52,21);if not (c2 in [#72,#80]) then h[i]:=ta;
            normvideo;if h[i]<=0 then write(#7);
            until h[i]>0
        end;
        restitue(20,20,2,36);
        prendre(20,8,2,47);
        gotoxy(10,21);
    end;
end;

```

```

write('Entrer le nombre de points à déterminer: ');
repeat
  gotoxy(50,21);write(' ');
  lire(51,21);if not (c2 in [#72,#80]) then np:=trunc(ta);
  if not (np in [1..5]) then
    begin
      write(#7);gotoxy(20,19);
      write('5 points au maximum ');delay(1000);
    end;
    gotoxy(20,19);clreol;
until np in [1..5];
restitue(20,8,2,47);
prendre(20,20,2,36);
gotoxy(21,21);write('Profondeur (cm) du point: ');
for i:=1 to np do
  begin
    gotoxy(47,21);write(i,':');
    textcolor(0);textbackground(7);
    repeat lire(50,21);if not (c2 in [#72,#80]) then
      zi[i]:=ta;
      if zi[i]<0 then write(#7);until zi[i]>=0;normvideo;
    end;
    restitue(20,20,2,36);
  h[nc]:=0;for i:=1 to nc-1 do h[nc]:=h[nc]+h[i];
  v:=zi[1];
  for i:=2 to np-1 do
    if zi[i]<v then v:=zi[i];
  if zi[np]<v then v:=zi[np];
  If (v<5)and(h[nc]>70) then
    Begin
      write(#7);gotoxy(10,3);clrscr;
      textcolor(7+blink);textbackground(red);
      write(' Superstructure trop grande ');
      normvideo;write('Fin des opérations');
      delay(5000);type_operat;
    End
  Else Begin
    prendre(20,20,2,51);
    gotoxy(21,21);
    write('Excentricité (cm) p/r à l''axe d''une roue: ');
    textcolor(0);textbackground(7);
    lire(64,21);if not (c2 in [#72,#80]) then r:=ta;
    normvideo;
    restitue(20,20,2,51);
    gotoxy(1,24);
    for k:=1 to 78 do write('-');
    gotoxy(15,25);
    write(' Esc-écran préc          F7-exéc          F10-Menu');
    gotoxy(75,25);
    repeat
      c:=readcar(bs);
    until c in [#27,#65,#68];

```

```

if c=#27 then select;
if c=#65 then
  begin
    textcolor(white);textbackground(red);
    prendre(18,20,3,45);
    gotoxy(22,19);write(' Inserez la disquette programme
dans');
    gotoxy(22,20);write(' l''unité A et appuyez une
touche');
    rev:=readkey;
    restitue(18,20,3,45);
    normvideo;
    clrscr;
    calculstruc;
  end;
if c=#68 then
  begin
    clrscr;
    type_operat;
  end
  End;
end;
clrscr;
End;          ( fin procedure select)

```

```

Procedure type_operat;
var
  mt,tou,rev:char;
  choi,a:int;
  bs:boolean;
Begin
  normvideo;clrscr;
  cadre2(8,20,18,53);
  gotoxy(22,10);write('CALCUL DE CONTRAINTES ');
  gotoxy(22,12);write('DIMENSIONNEMENT ');
  gotoxy(22,14);write('RAPPEL DE FICHER ');
  gotoxy(22,16);write('RETOUR AU SYSTEME DOS ');
  textcolor(blue);textbackground(19);
  gotoxy(21,5);write(' ':32);
  gotoxy(21,6);write(' ':32);
  gotoxy(30,6);write('MENU PRINCIPAL');
  gotoxy(21,7);write(' ':32);
  normvideo;
  gotoxy(23,22);
  write('Utiliser ',chr(24),' ',chr(25),' pour le choix. ');
  gotoxy(23,23);
  write('Et valider avec la touche "Entrée.");
  a:=10;
  tou:='b'; choi:=a;
  repeat
    if bs then
      Case tou of

```

```

#72: Begin
    If a>10 then a:=a-2
    Else if a=10 then a:=16;
    choi:=a;
End;
#80: Begin
    If a<16 then a:=a+2
    Else if a=16 then a:=10;
    choi:=a;
End;
End; (Case)
Case choi of
  10: begin
    textcolor(0);textbackground(7);
    gotoxy(22,10);write('CALCUL DE CONTRAINTES ');
    normvideo;
    gotoxy(22,12);write('DIMENSIONNEMENT ');
    gotoxy(22,16);write('RETOUR AU SYSTEME DOS ');
    gotoxy(28,10);write(chr(219));gotoxy(28,10);
  end;
  12: begin
    gotoxy(22,10);write('CALCUL DE CONTRAINTES ');
    textcolor(0);textbackground(7);
    gotoxy(22,12);write('DIMENSIONNEMENT ');
    normvideo;
    gotoxy(22,14);write('RAPPEL DE FICHER ');
    gotoxy(37,12);write(chr(219));gotoxy(37,12);
  end;
  14: begin
    gotoxy(22,12);write('DIMENSIONNEMENT ');
    textcolor(0);textbackground(7);
    gotoxy(22,14);write('RAPPEL DE FICHER ');
    normvideo;
    gotoxy(22,16);write('RETOUR AU SYSTEME DOS ');
    gotoxy(28,14);write(chr(219));gotoxy(28,14);
  end;
  16: begin
    textcolor(0);textbackground(7);
    gotoxy(22,16);write('RETOUR AU SYSTEME DOS ');
    normvideo;
    gotoxy(22,10);write('CALCUL DE CONTRAINTES ');
    gotoxy(22,14);write('RAPPEL DE FICHER ');
    gotoxy(28,16);write(chr(219));gotoxy(28,16);
  end;
End;
  rev:=readcar(bs); tou:=rev;
until (rev=#13)or(rev=#27);
gotoxy(15,22);clreol;
gotoxy(15,23);clreol;
Case rev of
#13: begin
    if a=10 then

```

```

begin
  operation:='calcul';
  cl:='n';select;
end;
if a=12 then
begin
  operation:='dimt';
  cl:='n';select;
end;
if a=14 then
begin
  operation:='rappel';
  nomfich;cl:='n';
  gotoxy(1,24);
  for k:=1 to 78 do write('-');
  normvideo;
  repeat
    textcolor(white);textbackground(red);
    gotoxy(1,25);clreol;
    gotoxy(15,25);
    write('          F4-impression
F10-Menu');
    gotoxy(75,25);normvideo;
    c:=readcar(bs);
    if c=#62 then
      begin
        gotoxy(1,25);clreol;
        gotoxy(15,24);
        write('Vérifiez l'imprimante et appuyez une
touche');
        mt:=readkey;
        impression;
      end;
    if c=#68 then type_operat;
  until (c=#68) and bs;
end;
if a=16 then
begin
  gotoxy(10,24);write('Sortie o/n ? ');
  cl:=upcase(readcar(bs)); write(cl);
  delay(100);gotoxy(10,24);clreol;
  if cl='O' then begin clrscr;halt end else
  type_operat;
end;
end;
#27: begin
  gotoxy(10,23);
  write('Etes-vous sûr de vouloir sortir du programme O/N
? ');
  cl:=upcase(readcar(bs));write(cl);delay(100);
  gotoxy(10,23);clreol;
  if cl<>'O' then type_operat else begin clrscr;halt;end

```

```
end;
End;      (type_operat)
End.
```

```

unit unite5;{Sortie des résultats}
interface
uses unite2,crt,printer;

procedure nomfich;
procedure affichage;
Procedure sauvegarde;
procedure impression;
Procedure selecthauteur;
procedure rappel_fichier;

      Implementation
Procedure cadre;
Begin
  for j:=1 to nc-1 do
    begin
      gotoxy(10+10*j,10);
      write('|| h',j,'=',h[j]:3:1);
    end;
  gotoxy(20,9);write(chr(201));
  gotoxy(20,11);write(chr(200));
  for j:=21 to 10+10*nc do
    begin
      gotoxy(j,9);write(chr(205));
      gotoxy(j,11);write(chr(205));
      if j mod 10=0 then
        begin
          gotoxy(j,9);write('⏏');
          gotoxy(j,11);write('⏏');
        end;
    end;
  gotoxy(10+10*nc,9);write(chr(187));
  gotoxy(10+10*nc,11);write(chr(188));
  gotoxy(10+10*nc,10);write(chr(186));
  gotoxy(25,10);
  textcolor(0+blink);textbackground(7);
  i:=1;write(h[i]:3:1);normvideo
End;

Procedure locale;
Begin
  gotoxy(15+10*i,10);
  textcolor(0+blink);textbackground(7);
  write(h[i]:3:1);gotoxy(15+10*i,10);
  normvideo;
End;

```

```

Procedure operation1;
Begin
  Case c of
    #75: Begin
      if i>1 then i:=i-1
      else if i=1 then i:=nc-1;
      locale
    End;
    #77: Begin
      if i<nc-1 then i:=i+1
      else if i=nc-1 then i:=1;
      locale
    End;
    #72: Begin
      h[i]:=h[i]+1;
      locale
    End;
    #80: Begin
      if h[i]>1 then
        begin
          h[i]:=h[i]-1;
          locale
        end
      else begin
        gotoxy(20,15);
        textcolor(0+blink);textbackground(7);
        write(#7);
        write('Valeur incompatible');
        c:=readkey;
        textbackground(0);gotoxy(20,15);
        writeln(' ');
        normvideo;
        gotoxy(16+6*i,10)
      end
    End
  End
End;
Procedure selecthauteur;
var
  bs:boolean;
Begin
  cadre;
  Repeat
    c:=readcar(bs);
    If (c in [#72,#75,#77,#80]) then
      Begin
        gotoxy(15+10*i,10);
        textcolor(7);textbackground(0);
        write(h[i]:3:1);
        normvideo;
        operation1;
      End;
    End;
  End;

```



```

Until c=#13;
clrscr
End;
Procedure sauvegarde;
Var
  fichier_donnee: file of tdonnee;
  donne_valeur:tdonnee;
  kc,kp,n_po:int;
  bs:boolean;
  mt:char;
Begin
  With donne_valeur do
    Begin
      For kc:=1 to nc do
        Begin
          t1[kc,1]:=el[kc];
          t1[kc,2]:=h[kc];
          t1[kc,3]:=mm[kc];
          cycl[kc-1]:=cyc[kc-1];
        End;
      cycl[nc]:=cyc[nc];nc1:=nc;
      jumelage:=dual;c_it:=itf1;
      mol2:=mo;condi_int:=interf;
      ope:=operation;qoo:=qo;
      vl:=vm;r1:=r;cr1:=cr;
      If operation='dimt' then
        Begin
          ct1:=ct;
          For kp:=1 to 2 do
            Begin
              all[kp]:=al[kp];
              For kc:=0 to nc-1 do
                Begin
                  t2[kc,kp,1]:=fd1[kc,kp];
                  t2[kc,kp,2]:=fd2[kc,kp];
                  t2[kc,kp,3]:=fd3[kc,kp];
                  t2[kc,kp,4]:=fd4[kc,kp];
                End;
              Case condi_int of
                1: Begin
                    case c_it of
                      12:begin
                          tsup2[2,kp,3]:=fdsup3[2,kp];
                          tsup2[2,kp,4]:=fdsup4[2,kp];
                        end;
                      23:begin
                          tsup2[3,kp,3]:=fdsup3[3,kp];
                          tsup2[3,kp,4]:=fdsup4[3,kp];
                        end;
                    end;
                End;
              2: Begin

```

```

                                tsup2[2,kp,3]:=fdsup3[2,kp];
                                tsup2[2,kp,4]:=fdsup4[2,kp];
                                tsup2[3,kp,3]:=fdsup3[3,kp];
                                tsup2[3,kp,4]:=fdsup4[3,kp];
                                End;
                                End;
                                End; (kp)
                                End
Else
Begin
    For kc:=1 to np do
        t1[kc,4]:=zi[kc];
        npl:=np;
        For n_po:=1 to np do
            Begin
                t3[n_po,1]:=f1[n_po];
                t3[n_po,2]:=f2[n_po];
                t3[n_po,3]:=f3[n_po];
                t3[n_po,4]:=f4[n_po];
            End;
            If interf<>0 then
                Begin
                    for n_po:=1 to np do
                        begin
                            tsup2[n_po,1,3]:=fsup3[n_po];
                            tsup2[n_po,1,4]:=fsup4[n_po];
                        end
                    End;
                End;
            End; ( with)
Assign(fichier_donnee,nom_fichier);
($i-)
rewrite(fichier_donnee);
($i+)
If ioresult=0 then
    Begin
        write(fichier_donnee,donne_valeur);
        close(fichier_donnee)
    End
Else
    Begin
        prendre(20,20,2,50);
        gotoxy(21,21);
        write('Ouverture du fichier ',mau,' impossible');
        delay(1000);
        restitue(20,20,2,50);
        if mt=#27 then exit;
    End;
End;

Procedure rappel_fichier;
Var

```

```

fichier_donnee: file of tdonnee;
donne_valeur:tdonnee;
kc,kp,n_po:int;
bs:boolean;
mt:char;
Begin
Assign(fichier_donnee,nom_fichier);
{$i-}
reset(fichier_donnee);
{$i+}
If ioresult=0 then
  Begin
    read(fichier_donnee,donne_valeur);
    With donne_valeur do
      Begin
        nc:=nc1;np:=np1;
        For kc:=1 to nc do
          Begin
            el[kc]:=t1[kc,1];
            h[kc]:=t1[kc,2];
            mm[kc]:=t1[kc,3];
            cyc[kc-1]:=cycl[kc-1];
          End;
        cyc[nc]:=cycl[nc];
        dual:=jumelage;itf1:=c_it;r:=r1;
        qo:=qoo;vm:=v1; operation:=ope;
        mo:=mol2;interf:=condi_int;
        cr:=cr1;mdl[1]:=el[1];
        If ope='dimt' then
          Begin
            ct:=ct1;
            For kp:=1 to 2 do
              Begin
                al[kp]:=a11[kp];
                For kc:=0 to nc-1 do
                  Begin
                    fd1[kc,kp]:=t2[kc,kp,1];
                    fd2[kc,kp]:=t2[kc,kp,2];
                    fd3[kc,kp]:=t2[kc,kp,3];
                    fd4[kc,kp]:=t2[kc,kp,4];
                  End;
                Case interf of
                1:Begin
                    case c_it of
                    12:begin
                        fdsup3[2,kp]:=tsup2[2,kp,3];
                        fdsup4[2,kp]:=tsup2[2,kp,4];
                    end;
                    23:begin
                        fdsup3[3,kp]:=tsup2[3,kp,3];
                        fdsup4[3,kp]:=tsup2[3,kp,4];
                    end;
                end;
              End;
            End;
          End;
        End;
      End;
    End;
  End;

```

```

        end;
    End;
    2:Begin
        fdsup3[2,kp]:=tsup2[2,kp,3];
        fdsup4[2,kp]:=tsup2[2,kp,4];
        fdsup3[3,kp]:=tsup2[3,kp,3];
        fdsup4[3,kp]:=tsup2[3,kp,4];
    End;
    End;
End; {kp}
End
Else
Begin
    For kc:=1 to np do
        zi[kc]:=t1[kc,4];
    For n_po:=1 to np do
        Begin
            f1[n_po]:=t3[n_po,1];
            f2[n_po]:=t3[n_po,2];
            f3[n_po]:=t3[n_po,3];
            f4[n_po]:=t3[n_po,4];
        End;
    If condi_int <> 0 then
        Begin
            for n_po:=1 to np do
                begin
                    tsup2[n_po,1,3]:=fsup3[n_po];
                    tsup2[n_po,1,4]:=fsup4[n_po];
                end
            End;
        End
    End;
    End;
    ( with)
    close(fichier_donnee)
End
Else
Begin
    prendre(20,20,2,50);
    gotoxy(21,21);
    write('Ouverture du fichier ',mau,' impossible');
    delay(1000);
    restitue(20,20,2,50);
    if c=#27 then exit;
End;
affichage;
End;

Function NomCorrect(nom:string):integer;
Var fich:text;
Begin
    NomCorrect:=10;
    Assign(fich,Nom);
    {$i-} reset(fich); {$i+}

```

```

    if Ioresult=0 then
      Begin
        NomCorrect:=0; close(fich);
      end;
    End;

Procedure nomfich;
var bs:boolean; Rev:char; Nf:integer;
Begin
  If operation='rappel' then
    Begin
      Repeat
        Rev:='0';
        prendre(20,50,2,28);
        gotoxy(51,21);write(' NOM DU FICHER: ');
        gotoxy(57,20);write('[unité]:[nom]');
        gotoxy(68,21);
        readln(nom_fichier);
        restitue(20,50,2,28);
        mau:=nom_fichier;
        if Pos('.',Mau)=0 Then
          nom_fichier:=Mau+'.vic' else
            Begin
              Delete(mau,Pos('.',Mau),Length(Mau)-Pos('.',Mau)+1);
              nom_fichier:=mau+'.vic';
            End;
        Nf:=NomCorrect(Nom_fichier);
        If Nf=0 then
          Begin
            mau:=nom_fichier;
            Rappel_fichier;
          End
        Else
          Begin
            prendre(20,10,3,20);
            gotoxy(11,21);write('Fichier inexistant');
            gotoxy(11,22);write('Reprise o/n ?');
            rev:=upcase(readkey);
            gotoxy(26,22);write(rev);delay(100);
            restitue(20,10,3,20);
          End
        Until ((Nf=0) or (Rev='N'));
      End
    End
  Else
    Begin
      prendre(20,50,2,28);
      gotoxy(51,21);write(' NOM DU FICHER: ');
      gotoxy(57,20);write('[unité]:[nom]');
      gotoxy(68,21);
      readln(nom_fichier);
      Mau:=Nom_fichier;
      restitue(20,50,2,28);
    End
  End
End

```

```

if Pos('.',Mau)=0 Then
nom_fichier:=Mau+'.vic' else
Begin
Delete(mau,Pos('.',Mau),Length(Mau)-Pos('.',Mau)+1);
nom_fichier:=mau+'.vic';
End;
Nf:=NomCorrect(Nom_fichier);
If Nf<>0 then Sauvegarde
Else
Begin
prendre(20,10,3,20);
gotoxy(11,21);write('Fichier existant');
gotoxy(11,22);write('Remplacer o/n ?');
rev:=upcase(readkey);
gotoxy(26,22);write(rev);delay(100);
restitue(20,10,3,20);
if Rev='O' then sauvegarde;
End;
End
End;

Procedure affichage;
Var
bs,bi:boolean;
kl,ki:int;
Begin
clrscr;bs:=true;
gotoxy(20,3);write('Nombre de couches: ',nc);
if interf=0 then writeln(' avec adhérence des couches');
if interf=1 then
if itf1=12 then writeln(' Interface 1 glissante')
else writeln(' Interface 2 glissante');
if interf=2 then writeln(' Interfaces 1 et 2 glissantes');
If operation='dimt' then
Begin
gotoxy(30,1);writeln(' DIMENSIONNEMENT');
el[1]:=mdl[1];gotoxy(10,4);
writeln('Coef-Pois      Modules (bars)      Epaisseurs (cm) ');
for i:=1 to nc do
begin
gotoxy(10,4+i);
write(' Mu',i,'= ',mm[i]:4:2,' E',i,'=',el[i]:5:0);
if i<=nc-1 then begin
gotoxy(43,4+i);writeln(' h',i,'=',h[i]:4:1);
end;
end;
writeln;
gotoxy(10,6+nc);writeln('Période 2: E1=',mo:5:0, 'bars ');
if ct='O' then
writeln(' Contrainte rupt couche de base ',cr:5:3, ' bars' );
gotoxy(6,8+nc);
if dual='O' then writeln('Roues jumelées - Charge:

```

```

',qo/0.5093:5:1,'t')
else writeln('Roue simple - Charge: ',qo/1.0186:5:1,'t');
gotoxy(6,9+nc);
writeln('Ray emp =12.5cm . Vitesse moyenne: ',vm*3.6:4:0,'
km/h');
for ki:=1 to 2 do
begin
gotoxy(10,10+nc); writeln(' Contraintes (bars) Période: ',ki);
gotoxy(6,11+nc);
writeln('Interface Verticale Radiale Tangent.
Cisaillement');
for i:=0 to nc-1 do
begin
gotoxy(6,12+i+nc);
write(' ',i,' ',qo*fd1[i,ki]:6:4);
gotoxy(32,12+i+nc);write(qo*fd3[i,ki]:6:4);
gotoxy(44,12+i+nc);write(qo*fd4[i,ki]:6:4);
gotoxy(55,12+i+nc);writeln(qo*fd2[i,ki]:6:4);
end;
gotoxy(20,24);write(' Appuyer une touche quelconque ');
c:=readkey>window(1,10+nc,80,25);
clrscr;delay(500);window(1,1,80,25);
end;
gotoxy(6,16);
write(al[1]:4:2,' de trafic en période froide et ');
writeln(al[2]:4:2,' en période chaude');
gotoxy(20,17);writeln(' Nombres de cycles admissibles ');
gotoxy(10,18);write(' Norniérage = ',cyc[0]:6);
gotoxy(10,19);writeln(' Nfissuration_enrobé = ',cyc[1]:6);
if ct='0' then
begin
gotoxy(10,20);write(' Nflexion couche traitée =
',cyc[2]:6);
for i:=3 to nc do
begin
gotoxy(10,18+i);
write(' Ncouche ',i,' :(poinçonnement)= ',cyc[i]:6);
end;
end
else for i:=2 to nc do
begin
gotoxy(10,18+i);
write(' Nsol non traité ',i,' :(poinçonnement)=
',cyc[i]:6);
end;
ny:=cyc[0];
for i:=1 to nc do
if ny>cyc[i] then ny:=cyc[i];
gotoxy(1,19+nc);
write(' Le nombre de cycles admissible est alors N=
',ny:8);
End

```

```

Else
  Begin
    gotoxy(30,1);write(' Calcul de contraintes à r =');
    if dual='0' then writeln(abs(37.5-r):5:2)
    else writeln(r:5:2);
    gotoxy(10,5);
    writeln('Coef-Pois      Modules (bars)      Epaisseurs (cm) ');
    for i:=1 to nc do
      begin
        gotoxy(10,5+i);
        write(' Mu',i,'= ',mm[i]:4:2,'      E',i,'=',el[i]:5:0);
        if i<=nc-1 then begin
          gotoxy(43,5+i);writeln(' h',i,'=',h[i]:4:1);
        end;
      end;
    gotoxy(6,7+nc);
    if dual='0' then
      writeln('Roues jumelées - Charge: ',qo/0.5093:4:1,' t')
    else writeln('Roue simple - Charge: ',qo/1.0186:4:1,' t');
    gotoxy(6,9+nc);writeln('Ray emp =12.5cm');
    gotoxy(20,10+nc); writeln(' Contraintes (bars) ');
    gotoxy(6,11+nc);
    writeln('Prof (cm)      Verticale      Radiale      Tangent.
    Cisaillement');
    for i:=1 to np do
      begin
        gotoxy(6,11+i+nc);
        write(' ',zi[i]:3:1);gotoxy(18,11+i+nc);write(qo*f1[i]:6:4);
        gotoxy(33,11+i+nc);write(qo*f3[i]:6:4);
        gotoxy(45,11+i+nc);write(qo*f4[i]:6:4);gotoxy(57,11+i+nc);
        writeln(qo*f2[i]:6:4);
      end;
    If nc>=2 then
      Begin
        case interf of
          1:begin
            if np>nc then gotoxy(6,12+2*np) else gotoxy(6,12+2*np);
            case itf1 of
              12:begin
                if operation='calcul' then
                  begin
                    kl:=0;
                    repeat
                      kl:=kl+1;
                      if zi[kl]=h[1] then bs:=true
                      else bs:=false;
                    until (bs=true) or (kl=np);
                    if bs=true then
                      begin
                        write('contradsup de la couche 2: ',qo*fsup3[kl]:6:4);
                        writeln('conttansup de la couche 2: ',qo*fsup4[kl]:6:4);
                      end
                    end
              end
            end
          end
        end
      end
    end
  end

```



```

        end
    else begin
        for ki:=1 to 2 do
            begin
write('contradsup de la couche 2: ',qo*fdsup3[1,ki]:6:4);
writeln(' conttansup de la couche 2: ',qo*fdsup4[1,ki]:6:4);
            end;
        end
    end;
23:begin
    if nc>=3 then
        begin
            if operation='calcul' then
                begin
                    kl:=0;
                    repeat
                        kl:=kl+1;
                        if zi[kl]=h[1]+h[2] then bs:=true
                        else bs:=false;
                    until (bs=true) or (kl=np);;
                    if bs=true then
                        begin
                            write('contradsup de la couche 3: ',qo*fsup3[kl]:6:4);
                            writeln(' conttansup de la couche 3: ',qo*fsup4[kl]:6:4);
                        end
                    end
                else begin
                    for ki:=1 to 2 do
                        begin
write('contradsup de la couche 3: ',qo*fdsup3[2,ki]:6:4);
writeln('conttansup de la couche 3: ',qo*fdsup4[2,ki]:6:4);
                        end;
                    end
                end;
            end;
        end;
    end;
2:begin
    if operation='calcul' then
        begin
            kl:=0;
            repeat
                kl:=kl+1;
if ((zi[kl]=h[1])and(nc>=2))or((zi[kl]= h[1]+h[2])and(nc>=3))
                then bs:=true else bs:=false;
            until (bs=true) or (kl=np);;
            if bs=true then
                begin
                    gotoxy(6,12+2*np);
                    write('contradsup de la couche 2: ',qo*fsup3[kl]:6:4);
                    writeln(' conttansup de la couche 2: ',qo*fsup4[kl]:6:4);
                    gotoxy(6,13+2*np);
                end
            end
        end
    end
end

```

```

write('contradsup de la couche 3: ',qo*fsup3[k1]:6:4);
writeln(' conttansup de la couche 3: ',qo*fsup4[k1]:6:4);
end;
end
else
begin
for ki:=1 to 2 do
begin
gotoxy(6,12+2*nc);
write('contradsup de la couche 2: ',qo*fdsup3[1,ki]:6:4);
writeln(' conttansup de la couche 2: ',qo*fdsup4[1,ki]:6:4);
gotoxy(6,13+2*nc);
if nc>=3 then begin
write('contradsup de la couche 3: ',qo*fdsup3[2,ki]:6:4);
writeln(' conttansup de la couche 3: ',qo*fdsup4[2,ki]:6:4);
end;
end;
end;
end;
end;
End;
End;
End;

```

```

Procedure deplace(x1,y1:byte;cas:string);

```

```

var ki: byte;

```

```

Begin

```

```

for ki:=x1 to y1 do

```

```

write(lst,cas);

```

```

End;

```

```

Procedure impression;

```

```

var

```

```

ki:byte;

```

```

bi,bs:boolean;

```

```

Begin

```

```

for i:=1 to 5 do

```

```

writeln(lst);

```

```

writeln(lst,'----- ');

```

```

writeln(lst,'! LOGICIEL ZOVROUTE !');

```

```

writeln(lst,'! Version 5.0 91 E.P.Thies !');

```

```

writeln(lst,'-----');

```

```

writeln(lst);

```

```

deplace(1,15,' ');

```

```

write(lst,'Nombre de couches: ',nc);

```

```

if interf = 0 then writeln(lst,' avec adherence des couches');

```

```

if interf=1 then

```

```

if itf1=12 then writeln(lst,' Interface 1 glissante')

```

```

else writeln(lst,' Interface 2 glissante');

```

```

if interf=2 then writeln(lst,' Interfaces 1 et 2 glissantes');

```

```

writeln(lst);

```

```

deplace(1,30,' ');

```

```

If operation='dimt' then
  Begin
    writeln(1st,'DIMENSIONNEMENT');
    writeln(1st);
    el[1]:=md1[1];
    writeln(1st,'                               Periode 1');
    deplace(1,20,' ');
    writeln(1st,'Coef-Pois           Modules (bars) ');
    writeln(1st);
    for i:=1 to nc do
      begin
        deplace(1,20,' ');
        writeln(1st,' Mu',i,'= ',mm[i]:4:2,'
          E',i,'=',el[i]:5:0);
        end;
      writeln(1st);
    deplace(1,20,' ');writeln(1st,'Periode 2:  E1=',mo:5:0,'bars ');
    writeln(1st);
    if ct='0' then
      deplace(1,15,' ');
      writeln(1st,'Contrainte de rupture couche de base ',cr:5:3,'
        bars' );
      writeln(1st);deplace(1,15,' ');
      if dual='0' then
        writeln(1st,'Ess.Std: Roues jumelees - Charge:
          ',qo/0.5093:5:1,' t');
      else writeln(1st,'Ess.Std: Roue simple - Charge:
        ',qo/1.0186:5:1,' t');
      writeln(1st);
      deplace(1,15,' ');
      writeln(1st,'Ray emp = 12.5 cm  Vitesse moyenne:
        ',vm*3.6:4:0,' km/h');
      writeln(1st);
      for ki:=1 to 2 do
        begin
          deplace(1,20,' '); writeln(1st,'  Contraintes (bars)
            Periode: ',ki);
          deplace(5,75,'-');writeln(1st);deplace(1,5,' ');
          write(1st,'Interface Verticale           Radiale
            Tangent. ');
          writeln(1st,'      Cisaillement');
          deplace(5,75,'-');writeln(1st);
        end;
      for i:=0 to nc-1 do
        begin
          deplace(1,6,' ');
          write(1st,' ',i,'           ',qo*fd1[i,ki]:8:4);
          deplace(23,28,' ');
          write(1st,qo*fd3[i,ki]:8:4);
          deplace(37,42,' ');
          write(1st,qo*fd4[i,ki]:8:4);
          deplace(51,56,' ');
          writeln(1st,qo*fd2[i,ki]:8:4);
        end;
    end;
  end;

```

```

end;                                     (i)
writeln(lst);
end;                                     (ki)
writeln(lst);
deplace(1,30,' ');writeln(lst,'Epaisseurs des couches');
writeln(lst); deplace(1,15,' ');
for i:=1 to nc-1 do
begin
deplace(16+6*i,21+6*i,' ');
write(lst,'h',i,'=',h[i]:5:2);
end;
writeln(lst);writeln(lst);
deplace(1,10,' ');
write(lst,100*a1[1]:3:0,'% de trafic en periode 1 (froide) et ');
writeln(lst,100*a1[2]:3:0,'% en periode 2 (chaude)');
writeln(lst);
for ki:=1 to 12 do
writeln(lst);
writeln(lst);
deplace(1,20,' ');writeln(lst,' Nombres de cycles admissibles ');
deplace(5,75,'-');writeln(lst);writeln(lst);
deplace(1,15,' ');writeln(lst,'Au niveau de l''enrobe');
deplace(1,20,' ');writeln(lst,'Nornierage =',cyc[0]:8);
deplace(1,20,' ');writeln(lst,'Nfissuration_enrobe =',cyc[1]:8);
if ct='0' then
begin
deplace(1,15,' ');writeln(lst,'Couche de base traitee');
deplace(1,20,' ');writeln(lst,'Nfissuration =',cyc[2]:8);
deplace(1,15,' ');writeln(lst,'Couches non traitees
(poinçonnement)');
for i:=3 to nc do
begin
deplace(1,20,' ');writeln(lst,'Ncouche ',i,'=',cyc[i]:8);
end;
end
else
begin
deplace(1,15,' ');writeln(lst,'Couches non traitees
(poinçonnement)');
deplace(1,20,' ');
for i:=2 to nc do
begin
write(lst,'Nsol ',i,'=',cyc[i]:8);
deplace(20+8*i,23+8*i,' ');
end
end;
ny:=cyc[0];
for i:=1 to nc do
if ny>cyc[i] then ny:=cyc[i];
writeln(lst);writeln(lst);
deplace(1,15,' ');
writeln(lst,'Le nombre de cycles admissible est alors N=

```

```

    ',ny:8);
    writeln(lst);writeln(lst);
    writeln(lst);deplace(1,20,' ');
    writeln(lst,'* Interface 0 = surface ');writeln(lst);
End
Else
Begin
write(lst,'CALCUL DE CONTRAINTES a r =');
if dual='0' then writeln(lst,abs(37.5-r):5:2)
else writeln(lst,r:5:2);
deplace(1,20,' ');
writeln(lst,'Coef-Pois      Modules (bars)      Epaisseurs (cm) ');
    writeln(lst);
    for i:=1 to nc do
        begin
            deplace(1,20,' ');
            write(lst,' Mu',i,'= ',mm[i]:4:2,'
E',i,'=',el[i]:5:0);
            if i<=nc-1 then
                writeln(lst,'          h',i,'=',h[i]:5:1);
            end;
            writeln(lst);writeln(lst);
            deplace(1,10,' ');
            if dual='0' then
                writeln(lst,'Ess.Std: Roues jumelees - Charge:
',qo/0.5093:5:1,' t')
            else writeln(lst,'Ess.Std: Roue simple - Charge:
',qo/1.0186:5:1,' t');
            writeln(lst);
            deplace(1,30,' ');writeln(lst,'Ray emp = 12.5 cm');
            writeln(lst);
            deplace(1,20,' ');
            writeln(lst,' Contraintes (bars) aux points donnes ');
            deplace(5,75,'-');writeln(lst);deplace(1,5,' ');
            write(lst,'Prof (cm) Verticales      Radiales      Tangent. ');
            writeln(lst,'      Cisaillement');
            deplace(5,75,'-');writeln(lst);
            for i:=1 to np do
                begin
                    deplace(1,8,' ');
                    write(lst,zi[i]:4:1,'          ',qo*f1[i]:8:4);
                    deplace(24,29,' ');
                    write(lst,qo*f3[i]:8:4);
                    deplace(38,41,' ');
                    write(lst,qo*f4[i]:8:4);
                    deplace(50,55,' ');
                    writeln(lst,qo*f2[i]:8:4);
                end;
                writeln(lst);
            End;
            If nc>=2 then
                Begin

```

```

case interf of
  1:begin
    case itf1 of
      12:begin
        if operation='calcul' then
          begin
            kl:=0;
            repeat
              kl:=kl+1;
              if zi[kl]=h[1] then bs:=true
              else bs:=false;
            until (bs=true) or (kl=np);
            if bs=true then
              begin
                write(lst,'contradsup de la couche 2: ',qo*fsup3[kl]:6:4);
                writeln(lst,' conttansup de la couche 2:',qo*fsup4[kl]:6:4);
              end
            end
          else begin
            for ki:=1 to 2 do
              begin
                write(lst,'contradsup de la couche 2: ',qo*fdsup3[1,ki]:6:4);
                writeln(lst,' conttansup de la couche 2:',qo*fdsup4[1,ki]:6:4);
              end;
            end
          end;
        23:begin
          if operation='calcul' then
            begin
              kl:=0;
              repeat
                kl:=kl+1;
                if zi[kl]=h[1]+h[2] then bs:=true
                else bs:=false;
              until (bs=true) or (kl=np);
              if bs=true then
                begin
                  write(lst,'contradsup de la couche 3:
                    ',qo*fsup3[kl]:6:4);
                  writeln(lst,' conttansup de la couche 3:
                    ',qo*fsup4[kl]:6:4);
                end
              end
            else begin
              if nc>=3 then begin
                for ki:=1 to 2 do
                  begin
                    write(lst,'contradsup de la couche 3:
                      ',qo*fdsup3[2,ki]:6:4);
                    writeln(lst,'conttansup de la couche 3:
                      ',qo*fdsup4[2,ki]:6:4);
                  end;
                end
              end
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

```

                end;
            end
        end;
    end;
end;
2:begin
    if operation='calcul' then
        begin
            kl:=0;
            repeat
                kl:=kl+1;
                if (zi[kl]=h[1])or(zi[kl]= h[1]+h[2]) then
                    bs:=true
                else bs:=false;
            until (bs=true) or (kl=np);
            if bs=true then
                begin
                    write(lst,'contradsup de la couche 2:
                    ',qo*fsup3[kl]:6:4);
                    writeln(lst,' conttansup de la couche 2:
                    ',qo*fsup4[kl]:6:4);
                    if nc>=3 then begin
                        write(lst,'contradsup de la couche 3:
                        ',qo*fsup3[kl]:6:4);
                        writeln(lst,' conttansup de la couche 3:
                        ',qo*fsup4[kl]:6:4);
                    end;
                end;
            end
        else
            begin
                for ki:=1 to 2 do
                    begin
                        write(lst,'contradsup de la couche 2:',qo*fdsup3[1,ki]:6:4);
                        writeln(lst,' conttansup de la couche 2:',qo*fdsup4[1,ki]:6:4);
                        if nc>=3 then begin
                            write(lst,'contradsup de la couche 3:
                            ',qo*fdsup3[2,ki]:6:4);
                            writeln(lst,' conttansup de la couche 3:
                            ',qo*fdsup4[2,ki]:6:4);
                        end;
                    end;
                end;
            end;
        end;
    end;
    (interf)
End;
writeln(lst,'
Auteur: ZOUNMENO Victor. Ing-G.C');
End;
End.

```

```

Program zovroute;
  uses crt,unite2,unite4;
Procedure menu;
var
c:char;
i,j,k,k1,k2:byte;
Begin
  normvideo;clrscr;
  k1:=40; k2:=41;
  textcolor(yellow);textbackground(blue);
  for i:=1 to 80 do
  begin
    if i mod 2=0 then
    begin
      k:=k1;
      k1:=k1-1;
    end
    else
    begin
      k:=k2;
      k2:=k2+1;
    end;
    for j:=1 to 24 do
    begin
      gotoxy(k,j);write('█');
    end;
  end;
  window(10,9,70,19);
  clrscr;
  window(1,1,80,25);
  textcolor(yellow+blink);textbackground(blue);
  gotoxy(12,10);
  write('
█          █          █          █          █          █          █          █
█');
  for i:=1 to 2 do
  begin
    gotoxy(12,10+i);
    write('█          █          █          █          █          █          █          █');
    end;
    gotoxy(12,13);
    write('█          █          █          █          █          █          █          █');
  ');
  gotoxy(12,14);
  write('█          █          █          █          █          █          █          █');
  gotoxy(12,15);
  write('█          █          █          █          █          █          █          █');
  gotoxy(12,16);
  write('█          █          █          █          █          █          █          █');
  ');

```



```
normvideo;textcolor(yellow);textbackground(green);  
gotoxy(30,18);write('Version E.P.T 1991');  
normvideo;  
End;
```

```
Begin           {Programme principal}  
  el[1]:=50000;el[2]:=80000;el[3]:=5000;el[4]:=300;  
  mm[1]:=0.5;mm[2]:=0.5;mm[3]:=0.5;mm[4]:=0.5;  
  h[1]:=8;H[2]:=12;h[3]:=35;h[4]:=50;nc:=4;np:=4;  
  menu;delay(2000);  
  type_operat;  
  clrscr;  
End.
```

 ! LOGICIEL ZOVROUTE !
 ! Version 5.0 91 E.P.Thies !

Nombre de couches: 5 avec adherence des couches

CALCUL DE CONTRAINTES a r = 0.00
 Coef-Pois Modules (bars) Epaisseurs (cm)

Mu1= 0.4	E1=80000	h1= 10.0
Mu2= 0.4	E2=70000	h2= 15.0
Mu3= 0.3	E3=50000	h3= 20.0
Mu4= 0.3	E4= 5000	h4= 20.0
Mu5= 0.3	E5= 800	

Ess.Std: Roues jumelees - Charge: 19.0 t

Ray emp = 12.5 cm

Contraintes (bars) aux points donnes

Prof (cm)	Verticales	Radiales	Tangent	Cisaillt
0.0	-9.6683	-12.4013	-13.0334	-0.0000
10.0	-6.9542	-2.5673	-2.4260	-0.3999
15.0	-4.7924	-1.2318	-0.9479	-0.5275
25.0	-2.0150	0.9755	1.4212	-0.5697
65.0	-0.0911	0.2899	0.3363	-0.0179

 ! LOGICIEL ZOVROUTE !
 ! Version 5.0 91 E.P.Thies !

Nombre de couches: 3 Interface 1 glissante

CALCUL DE CONTRAINTES a r = 0.00
 Coef-Pois Modules (bars) Epaisseurs (cm)
 Mu1= 0.4 E1=50000 h1= 6.0
 Mu2= 0.3 E2=40000 h2= 15.0
 Mu3= 0.3 E3= 5000

Ess.Std: Roue simple - Charge: 6.5 t

Ray emp = 12.5 cm

Contraintes (bars) aux points donnes

Prof (cm)	Verticales	Radiales	Tangent	Cisaillt
0.0	-6.5945	-8.3457	-8.3457	0.0000
6.0	-6.4957	5.6363	5.6363	0.0000
21.0	-1.6827	5.0621	5.0621	0.0000
25.0	-1.3526	0.0230	0.0230	0.0000

contrad sup de la couche 2: -15.4535 conttansup de la couche 2: -15.4535

 ! LOGICIEL ZDROUTE !
 ! Version 5.0 91 E.P.Thies !

Nombre de couches: 4 avec adherence des couches

DIMENSIONNEMENT

Periode 1

Coef-Pois Modules (bars)

Mu1= 0.4 E1=80000
 Mu2= 0.3 E2=60000
 Mu3= 0.3 E3= 5000
 Mu4= 0.3 E4= 500

Periode 2: E1=50000bars

Contrainte de rupture couche de base 25.000 bars

Roues jumelees - Charge: 13.0 t

Ray emp = 12.5 cm Vitesse moyenne: 70 km/h

Contraintes (bars) Periode: 1

Interface	Verticales	Radiales	Tangent	Cisaillt
0	-6.6152	-19.1600	-21.5933	0.0000
1	-5.7512	-8.0291	-8.8986	-0.4759
2	-1.4093	8.8679	10.9596	-0.3067
3	-0.2255	1.0912	1.3242	-0.0444

Contraintes (bars) Periode: 2

Interface	Verticales	Radiales	Tangent	Cisaillt
0	-6.6152	-15.3294	-17.2036	0.0000
1	-5.8612	-8.1931	-9.0172	-0.3782
2	-1.5299	8.9359	11.0695	-0.3226
3	-0.2442	1.1688	1.4212	-0.0479

Epaisseurs des couches

h1 = 4.00 h2 =10.00 h3 =25.00

30% de trafic en periode froide et 70% en periode 2 (chaude)

Nombres de cycles admissibles

Au niveau de l'enrobe
Nornierage = 1.0E+20
Nfissur_enrobe = 3.4E+07
Couche de base traitee
Nft1 = 7.4E+12
Couches non traitees
Nsol 3= 5.3E+12
Nsol 4= 2.9E+07

Le nombre de cycles admissible est alors N= 29292625

* Interface 0 = surface

```

-----
!   LOGICIEL ZOVROUTE   !
!   Version 5.0 91   E.P.Thies   !
-----

```

Nombre de couches: 4 avec adherence des couches

DIMENSIONNEMENT

Periode 1

Coef-Pois

Modules (bars)

Mu1= 0.4

E1=50000

Mu2= 0.4

E2=50000

Mu3= 0.3

E3= 5000

Mu4= 0.3

E4= 800

Periode 2:

E1=40000bars

Contrainte de rupture couche de base 25.000 bars

Ess.Std: Roues jumelees - Charge: 13.0 t

Ray emp = 12.5 cm Vitesse moyenne: 70 km/h

Contraintes (bars) Periode: 1

Interface	Verticale	Radiale	Tangent.	Cisaillement
0	-6.6152	-15.3633	-17.2372	-0.0000
1	-5.8780	-6.6540	-7.3224	-0.3406
2	-1.6869	8.0459	9.6664	-0.2878
3	-0.2761	0.8095	0.9898	-0.0556

Contraintes (bars) Periode: 2

Interface	Verticale	Radiale	Tangent.	Cisaillement
0	-6.6152	-12.6052	-14.0716	-0.0000
1	-5.9634	-6.8013	-7.4248	-0.2701
2	-1.7902	8.0680	9.7131	-0.2964
3	-0.2920	0.8478	1.0396	-0.0588

Epaisseurs des couches

h1 = 4.00

h2 =10.00

h3 =30.00

30% de trafic en periode 1 (froide) et 70% en periode 2 (chaude)

Nombres de cycles admissibles.

Au niveau de l'enrobe

Normierage = $1.0E+20$

Nfissuration_enrobe = $9.4E+07$

Couche de base traitée

Nfissuration = $3.6E+13$

Couches non traitées (poinçonnement)

Ncouche 3= $4.6E+10$

Ncouche 4= $2.7E+10$

Le nombre de cycles admissible est alors $N= 9.4E+07$

* Interface 0 = surface

Auteur: ZDUNMENDOU Victor. Ing-G.C

```

-----
! LOGICIEL ZOVROUTE !
! Version 5.0 91 E.P.Thies !
-----

```

Nombre de couches: 4 avec adherence des couches

DIMENSIONNEMENT

Periode 1

Coef-Pois

Modules (bars)

```

Mu1= 0.4      E1=80000
Mu2= 0.3      E2=60000
Mu3= 0.3      E3= 5000
Mu4= 0.3      E4= 500

```

Periode 2: E1=50000bars

Contrainte de rupture couche de base 25.000 bars

Ess.Std: Roues jumlees - Charge: 13.0 t

Ray emp = 12.5 cm Vitesse moyenne: 70 km/h

Contraintes (bars) Periode: 1

Interface	Verticale	Radiale	Tangent.	Cisaillement
0	-6.6152	-18.2299	-20.5564	-0.0000
1	-5.7652	-7.6696	-8.4976	-0.4333
2	-1.4827	8.2215	10.2280	-0.2937
3	-0.1923	0.9481	1.1457	-0.0385

Contraintes (bars) Periode: 2

Interface	Verticale	Radiale	Tangent.	Cisaillement
0	-6.6152	-14.5696	-16.3559	-0.0000
1	-5.8733	-7.8458	-8.6291	-0.3407
2	-1.6037	8.2611	10.3031	-0.3068
3	-0.2069	1.0107	1.2243	-0.0414

Epaisseurs des couches

h1 = 4.00 h2 = 10.00 h3 = 30.00

30% de trafic en periode 1 (froide) et 70% en periode 2 (chaude)

Nombres de cycles admissibles

Au niveau de l'enrobe
Norniérage = $1.0E+20$
Nfissuration_enrobe = $5.3E+07$
Couche de base traitée
Nfissuration = $2.5E+13$
Couches non traitées (poinçonnement)
Ncouche 3= $1.2E+12$
Ncouche 4= $1.1E+09$

Le nombre de cycles admissible est alors N= $5.3E+07$

* Interface 0 = surface

Auteur: ZOUNHENDU Victor, Ing-B.C

 ! LOGICIEL ZDVRROUTE !
 ! Version 5.0 91 E.P.Thies !

Nombre de couches: 4 avec adherence des couches

DIMENSIONNEMENT

Periode 1

Coef-Pois Modules (bars)

Mu1= 0.4 E1=80000
 Mu2= 0.3 E2=60000
 Mu3= 0.3 E3= 5000
 Mu4= 0.3 E4= 500

Periode 2: E1=50000bars

Contrainte de rupture couche de base 25.000 bars

Roues jumelees - Charge: 13.0 t

Ray emp = 12.5 cm Vitesse moyenne: 70 km/h

Contraintes (bars) Periode: 1

Interface	Verticales	Radiales	Tangent	Cisaillt
0	-6.6152	-16.0791	-19.4101	-0.0000
1	-5.4752	-5.0400	-5.6288	-0.4776
2	-1.3570	7.8458	9.7637	-0.2793
3	-0.1802	0.8916	1.0749	-0.0360

Contraintes (bars) Periode: 2

Interface	Verticales	Radiales	Tangent	Cisaillt
0	-6.6152	-12.8321	-14.6391	-0.0000
1	-5.6458	-5.7447	-6.3814	-0.3876
2	-1.4865	7.9414	9.9033	-0.2935
3	-0.1957	0.9584	1.1583	-0.0390

Epaisseurs des couches

h1 = 5.00 h2 =10.00 h3 =30.00

30% de trafic en periode froide et 70% en periode 2 (chaude)

! LOGICIEL ZOVROUTE !
! Version 5.0 91 E.P.Thies !

Nombre de couches: 4 avec adherence des couches

CALCUL DE CONTRAINTES a r = 0.00

Coef-Pois	Modules (bars)	Epaisseurs (cm)
Mu1= 0.3	E1=10000	h1= 10.0
Mu2= 0.3	E2=10000	h2= 10.0
Mu3= 0.3	E3=10000	h3= 10.0
Mu4= 0.3	E4=10000	

Ess.Std: Roue simple - Charge: 6.5 t

Ray emp = 12.5 cm

Contraintes (bars) aux points donnes

Prof (cm)	Verticales	Radiales	Tangent	Cisailt
0.0	-6.5945	-5.2756	-5.2756	0.0000
10.0	-5.0065	-0.7272	-0.7272	0.0000
20.0	-2.5873	-0.0140	-0.0140	0.0000
30.0	-1.4257	0.0566	0.0566	0.0000

! LOGICIEL ZOVROUTE !
! Version 5.0 91 E.P.Thies !

Nombre de couches: 4 avec adherence des couches

CALCUL DE CONTRAINTES a r = 0.00

Coef-Pois	Modules (bars)	Epaisseurs (cm)
Mu1= 0.4	E1=50000	h1= 5.0
Mu2= 0.4	E2=40000	h2= 15.0
Mu3= 0.3	E3= 5000	h3= 30.0
Mu4= 0.3	E4= 500	

Ess.Std: Roue simple - Charge: 6.5 t

Ray emp = 12.5 cm

Contraintes (bars) aux points donnes

Prof (cm)	Verticales	Radiales	Tangent	Cisailt
6.3	-5.5310	-3.1090	-3.1090	0.0000
10.0	-4.0164	-0.6655	-0.6655	0.0000
20.0	-0.9500	5.2494	5.2493	0.0000
30.0	-0.4429	0.2255	0.2255	0.0000
50.0	-0.0728	0.5028	0.5028	0.0000

 ! LOGICIEL ZOVROUTE !
 ! Version 5.0 91 E.P.Thies !

Nombre de couches: 4 avec adherence des couches

CALCUL DE CONTRAINTES a r =18.75
 Coef-Pois Modules (bars) Epaisseurs (cm)
 Mu1= 0.4 E1=50000 h1= 5.0
 Mu2= 0.4 E2=40000 h2= 15.0
 Mu3= 0.3 E3= 5000 h3= 30.0
 Mu4= 0.3 E4= 500

Ess.Std: Roue simple - Charge: 6.5 t

Ray emp = 12.5 cm

Contraintes (bars) aux points donnes

Prof (cm)	Verticales	Radiales	Tangent	Cisaillt
5.0	-0.2204	-2.9910	-2.1732	-1.1472
6.3	-0.3111	-2.1449	-1.3916	-1.3236
10.0	-0.4921	-1.0654	-0.2770	-1.5302
20.0	-0.4542	1.8334	2.9927	-0.3616
50.0	-0.0988	0.5179	0.5341	-0.0228

Effet de μ .

! LOGICIEL ZOVROUTE !
! Version 5.1 91 E.P.Thies !

Nombre de couches: 4 avec adherence des couches

CALCUL DE CONTRAINTES a r = 0:00

Coef-Pois Modules (bars) Epaisseurs (cm)

Mu1= 0.3 E1=50000 h1= 5.0

Mu2= 0.3 E2=40000 h2= 15.0

Mu3= 0.3 E3= 5000 h3= 30.0

Mu4= 0.3 E4= 300

Roue simple - Charge: 6.5 t

Ray emp = 12.5 cm

Contraintes (bars) aux points donnes

Prof (cm)	Verticales	Radiales	Tangent	Cisailt
0.0	-6.5945	-10.1267	-10.1267	0.0000
5.0	-5.9333	-3.7827	-3.7827	0.0000
20.0	-0.9604	4.7792	4.7791	0.0000
50.0	-0.0479	0.5721	0.5721	0.0000

```

! LOGICIEL ZDROUTE !
! Version 5.0 91 E.P.Thies !

```

Nombre de couches: 4 avec adherence des couches

CALCUL DE CONTRAINTES a r = 0.00

Coef-Pois Modules (bars) Epaisseurs (cm)

Mu1= 0.4 E1=50000 h1= 5.0
 Mu2= 0.4 E2=40000 h2= 15.0
 Mu3= 0.3 E3= 5000 h3= 30.0
 Mu4= 0.3 E4= 500

Roue simple - Charge: 6.5 t

Ray emp = 12.5 cm

Contraintes (bars) aux points donnees

Prof (cm)	Verticales	Radiales	Tangent	Cisaillt
0.0	-6.5945	-11.1999	-11.1999	0.0000
5.0	-5.9396	-4.1631	-4.1630	0.0000
20.0	-0.9500	5.2494	5.2493	0.0000
50.0	-0.0728	0.5028	0.5028	0.0000

```

-----
!   LOGICIEL ZDROUTE   !
!   Version 5.0 91   E.P.Thies   !
-----

```

Nombre de couches: 4 avec adherence des couches

CALCUL DE CONTRAINTES a r =18.75

Coef-Pois Modules (bars) Epaisseurs (cm)

Mu1= 0.4 E1=50000 h1= 5.0
 Mu2= 0.4 E2=40000 h2= 15.0
 Mu3= 0.3 E3= 5000 h3= 30.0
 Mu4= 0.3 E4= 500

Roue simple - Charge: 6.5 t

Ray emp = 12.5 cm

Contraintes (bars) aux points donnees

Prof (cm)	Verticales	Radiales	Tangent	Cisailt
0.0	-0.1027	-3.0435	-4.3269	0.0000
5.0	-0.2204	-2.9910	-2.1732	-1.1472
20.0	-0.4542	1.8334	2.9927	-0.3616
50.0	-0.0988	0.5179	0.5341	-0.0228

 ! LOGICIEL ZOVROUTE !
 ! Version 5.0 91 E.P.Thies !

Nombre de couches: 4 avec adherence des couches

CALCUL DE CONTRAINTES a r = 37.5

Coef-Pois Modules (bars) Epaisseurs (cm)

Mu1= 0.4 E1=50000 h1= 5.0
 Mu2= 0.4 E2=40000 h2= 15.0
 Mu3= 0.3 E3= 5000 h3= 30.0
 Mu4= 0.3 E4= 500

Roue simple - Charge: 6.5 t

Ray emp = 12.5 cm

Contraintes (bars) aux points donnes

Prof (cm)	Verticales	Radiales	Tangent	Cisailt
0.0	-0.0207	-1.1340	-2.3641	-0.0000
5.0	-0.0188	-0.9037	-1.3248	-0.3507
20.0	-0.1326	0.1161	1.2343	-0.2459
50.0	-0.0765	0.2460	0.3970	-0.0299

```

-----
! LOGICIEL ZOVROUTE !
! Version 5.1 91 E.P.Thies !
-----

```

Nombre de couches: 4 avec adherence des couches

CALCUL DE CONTRAINTES a r = 12.5 cm

Coef-Pois Modules (bars) Epaisseurs (cm)

Mu1= 0.4 E1=50000 h1= 5.0
 Mu2= 0.4 E2=40000 h2= 15.0
 Mu3= 0.3 E3= 5000 h3= 30.0
 Mu4= 0.3 E4= 500

Roue simple - Charge: 6.5 t

Ray emp = 12.5 cm

Contraintes (bars) aux points donnees

Prof (cm)	Verticales	Radiales	Tangent	Disailit
3.0	-6.4005	-6.7440	-6.7440	0.0000
10.0	-4.0164	-0.6655	-0.6655	0.0000
30.0	-0.4429	0.2255	0.2255	0.0000
55.0	-0.0602	0.0186	0.0186	0.0000

```

-----
!   LOGICIEL ZOVROUTE   !
!   Version 5.0 91   E.P.Thies   !
-----

```

Nombre de couches: 4 avec adhérence des couches

CALCUL DE CONTRAINTES a r =18.75

Coef-Pois Modules (bars) Epaisseurs (cm)

Mu1= 0.4 E1=40000 h1= 5.0
 Mu2= 0.4 E2=50000 h2= 15.0
 Mu3= 0.3 E3= 5000 h3= 30.0
 Mu4= 0.3 E4= 500

Roue simple - Charge: 6.5 t

Ray emp = 12.5 cm

Contraintes (bars) aux points donnes

Prof (cm)	Verticales	Radiales	Tangent	Cisaillt
0.0	-0.1027	-2.8701	-3.6888	0.0000
5.0	-0.1690	-2.4499	-2.0347	-1.0540
20.0	-0.4412	2.2505	3.5569	-0.3360
50.0	-0.0988	0.5122	0.5264	-0.0222

```

-----
!   LOGICIEL ZOVROUTE   !
!   Version 5.0 91   E.P.Thies   !
-----

```

Nombre de couches: 4 avec adherence des couches

CALCUL DE CONTRAINTES a r =37.50

Coef-Pois Modules (bars) Epaisseurs (cm)

```

      Mu1= 0.4            E1=40000            h1= 5.0
      Mu2= 0.4            E2=50000            h2= 15.0
      Mu3= 0.3            E3= 5000            h3= 30.0
      Mu4= 0.3            E4= 500

```

Roue simple - Charge: 6.5 t

Ray emp = 12.5 cm

Contraintes (bars) aux points donnees

Prof (cm)	Verticales	Radiales	Tangent	Cisaillt
0.0	-0.0207	-0.9972	-2.0234	-0.0000
5.0	-0.0179	-0.7360	-1.1817	-0.3149
20.0	-0.1381	0.2079	1.4969	-0.2379
50.0	-0.0769	0.2472	0.3939	-0.0295

```

-----
!   LOGICIEL ZOVRROUTE   !
!   Version 5.0 91   E.P.Thies   !
-----

```

Nombre de couches: 4 avec adherence des couches

CALCUL DE CONTRAINTES a r = 0.00

Coef-Pois Modules (bars) Epaisseurs (cm)

Mu1= 0.4 E1=40000 h1= 5.0

Mu2= 0.4 E2=50000 h2= 15.0

Mu3= 0.3 E3= 5000 h3= 30.0

Mu4= 0.3 E4= 500

Roue simple - Charge: 6.5 t

Ray emp = 12.5 cm

Contraintes (bars) aux points donnes

Prof (cm)	Verticales	Radiales	Tangent	Cisailt
0.0	-6.5945	-9.5498	-9.5498	0.0000
5.0	-6.0128	-5.0068	-5.0068	0.0000
20.0	-0.8902	6.1562	6.1561	0.0000
50.0	-0.0715	0.4899	0.4899	0.0000

```

-----
! LOGICIEL ZOVROUTE !
! Version 5.0 91 E.P.Thies !
-----

```

Nombre de couches: 4 avec adherence des couches

CALCUL DE CONTRAINTES a r = 0.00

Coef-Pois Modules (bars) Epaisseurs (cm)

```

      Mu1= 0.4            E1=80000            h1= 5.0
      Mu2= 0.3            E2=30000            h2= 15.0
      Mu3= 0.3            E3= 5000            h3= 30.0
      Mu4= 0.3            E4= 500

```

Roue simple - Charge: 6.5 t

Ray emp = 12.5 cm

Contraintes (bars) aux points donnees

Prof (cm)	Verticales	Radiales	Tangent	Cisailt
0.0	-6.5945	-14.5377	-14.5377	0.0000
5.0	-5.6750	-1.3925	-1.3924	0.0000
20.0	-0.9976	3.6844	3.6844	0.0000
50.0	-0.0724	0.5078	0.5078	0.0000

Nombres de cycles admissibles

Au niveau de l'enrobe
Nornierage = $1.4E+16$
Nfissur_enrobe = $2.1E+08$
Couche de base traitee
Nft1 = $4.7E+13$
Couches non traitees
Nsol 3= $1.4E+13$
Nsol 4= $4.5E+09$

Le nombre de cycles admissible est alors $N = 2,1 \cdot 10^8$

* Interface 0 = surface