



HAL
open science

Un processeur et un logiciel de resolution sur Mac Intosh pour des problemes de structure de poutres en dimension deux

C. Compe, O. Michaux, B. Muller

► **To cite this version:**

C. Compe, O. Michaux, B. Muller. Un processeur et un logiciel de resolution sur Mac Intosh pour des problemes de structure de poutres en dimension deux. RT-0074, INRIA. 1986, pp.32. inria-00070086

HAL Id: inria-00070086

<https://inria.hal.science/inria-00070086>

Submitted on 19 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

IRIA

CENTRE DE ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél: (1) 39 63 55 11

Rapports Techniques

N° 74

UN PROCESSEUR ET UN LOGICIEL DE RÉOLUTION SUR MAC INTOSH POUR DES PROBLÈMES DE STRUCTURE DE POUTRES EN DIMENSION DEUX

Catherine COMPE
Olivier MICHAUX
Bruno MULLER

Septembre 1986

UN PREPROCESSEUR ET UN LOGICIEL DE RESOLUTION SUR MAC INTOSH
POUR DES PROBLEMES DE STRUCTURES DE POUTRES EN DIMENSION DEUX.

A PRE-PROCESSOR AND A SOLVER OF IN-PLANE STRUCTURAL PROBLEMS
WITH BEAMS FOR MAC INTOSH

Catherine COMPE

Olivier MICHAUX

Bruno MULLER

Septembre 1986



PAPIER RECUPERÉ ET RECYCLÉ

**UN PREPROCESSEUR ET UN LOGICIEL DE RESOLUTION SUR MAC INTOSH
POUR DES PROBLEMES DE STRUCTURES DE POUTRES EN DIMENSION DEUX**

Résumé

Ce document décrit l'élaboration, la mise en oeuvre et l'utilisation de l'application Treillis sur Mac Intosh. Ce logiciel permet de saisir la géométrie et les caractéristiques physiques de tout treillis de poutres en dimension deux, de générer les structures de données propres au code d'éléments finis Modulef, de résoudre le problème, de restituer les déformations.

Après une brève description logicielle et matérielle du Mac Intosh on trouve un guide pour l'utilisateur donnant le détail des fonctionnalités offertes pour l'application : l'aspect saisie clavier et saisie graphique, la gestion des fichiers et les modes de résolution. Le dernier chapitre est un guide pour le développeur destiné à faciliter la maintenance. Il présente l'architecture modulaire du logiciel, la structure du fichier et un algorithme général du programme.

Abstract

This document describes the elaboration, the implementation and the use of the Treillis application in the Mac Intosh. The software makes possible to insert the geometry and the physical characteristics of each two-sized beam treillis-work, to generate the data structures incident to the Modulef finite elements code, to solve the problem and to give back the deformations.

After a short description software and hardware of the Mac Instosh, we find the booklet for the user, giving the details of the functions offered for the application : the keyboard entry aspect and the graphic one, the files maintenance and resolution modes. The last chapter is a programmer's booklet intended to facilitate the maintenance. It presents the modular architecture of the software, the files structure and the general algorithm of the program.

SOMMAIRE

I. Introduction.	<i>page 1</i>
II. Mise en oeuvre sur Mac Intosh.	<i>page 2</i>
2.1) Présentation du matériel utilisé.	<i>page 2</i>
2.1.1) L'aspect hardware.	<i>page 2</i>
2.1.2) L'aspect software.	
a) Les Managers.	<i>page 2</i>
b) Les ressources.	<i>page 5</i>
2.2) Choix du langage Fortran et spécifications.	<i>page 6</i>
2.2.1) Choix du langage Fortran.	<i>page 6</i>
2.2.2) Spécifications.	<i>page 6</i>
a) Les données à manipuler.	<i>page 6</i>
b) Les fonctionnalités désirées.	<i>page 7</i>
III. Description externe ou guide pour l'utilisateur.	<i>page 8</i>
3.1) Généralités.	<i>page 8</i>
3.2) Le menu principal et les diverses fonctionnalités.	<i>page 9</i>
3.3) L'aspect saisie graphique : le menu "Tables".	<i>page 10</i>
3.4) L'aspect saisie graphique : le menu "Géométrie".	<i>page 16</i>
3.5) La fonction résolution.	<i>page 19</i>
3.6) La fonction fichier.	<i>page 22</i>
IV. Description interne ou guide pour le développeur.	<i>page 24</i>
4.1) Architecture générale des différents modules de l'application.	<i>page 24</i>
4.2) Les contraintes de programmation sous Mac Intosh.	<i>page 25</i>
4.3) La structure des données.	<i>page 25</i>
4.4) La structure des fichiers.	<i>page 28</i>
4.5) L'algorithme du programme principal.	<i>page 30</i>
V. Conclusion.	

I. INTRODUCTION.

Lors de l'étude numérique d'une structure (une aile d'avion, un bateau, une plaque, une coque, une barre...), il est nécessaire de savoir si elle répond aux critères de solidité, de légèreté, (etc...) attendus.

Il faut notamment examiner les déformations et les réactions de cette structure sous l'effet de contraintes : charges, pression... ,et localiser ainsi les différents points "fragiles".

La méthode des éléments finis, par une modélisation en équations aux dérivées partielles, est un outil industriel utilisé pour l'étude de ces problèmes.

A l'heure actuelle, les logiciels d'éléments finis demandent une saisie plus ou moins laborieuse des données. Par exemple, pour définir correctement la structure d'un avion, il faut au moins 30000 points. La saisie à elle seule est phénoménale. D'où l'importance d'un préprocesseur et d'un postprocesseur interactifs très "conviviaux" qui traitent les entrées/sorties des différentes données ainsi que la visualisation graphique de la structure.

L'application *TREILLIS* s'insère dans le cadre spécifique de l'élaboration d'un préprocesseur ergonomique pour le code d'Element Finis *Modulef*.

Elle permettra :

- *De saisir* la géométrie et les caractéristiques physiques de tout treillis de poutres en dimension deux.
- *De générer* les structures de données nécessaires à *Modulef* pour la résolution du problème.
- *De restituer* les déformations subies par ce treillis.

Elle se limitera au cas de l'étude d'un problème statique et nous tenterons de prouver le bien fondé d'une telle application pour des ingénieurs en calcul scientifique.

Le choix du matériel a été essentiellement guidé par le souci d'ergonomie, de fiabilité et de transportabilité du logiciel. A l'heure actuelle, le *Mac Intosh* est un micro-ordinateur qui satisfait ces trois critères. Il a été conçu aussi bien pour des applications graphiques que pour du traitement de texte, ce qui lui confère une qualité supplémentaire : il est très répandu.

II. MISE EN OEUVRE SUR MAC INTOSH.

Ce chapitre n'a pas la prétention de donner une présentation détaillée du Mac Intosh aussi bien au niveau hardware qu'au niveau software. Il est nécessaire, néanmoins, pour utiliser l'application Treillis, de manière rationnelle, de présenter quelques unes des caractéristiques techniques de ce micro_ordinateur, la spécificité de chacune de ses bibliothèques ainsi que le concept de fichier ressource qui est très utilisé. Les bibliothèques regroupent 470 routines, résidentes en R.O.M.(Read Only Memory), et sont des outils de base indispensables à tout développeur.

2.1) Présentation du matériel utilisé.

2.1.1) L' Aspect hardware

Le Mac Intosh se compose :

- d'un micro processeur MC68 000, 128 ou 512 K octets de mémoire centrale
- d'un lecteur de disquettes intégré, avec possibilité d'extension d'un deuxième lecteur .
- d'un écran graphique haute résolution (512 X 342 pixels).

2.1.2) L' Aspect Software.

Le Mac Intosh est fait pour les applications graphiques. Il possède en R.O.M.(Read Only Memory) des bibliothèques impressionnantes appelées Managers et regroupant 470 routines graphiques; elles permettent la gestion des applications.

a) LES MANAGERS.

Les routines graphiques du Mac Intosh sont des routines de base : elles ont été créées pour faciliter la manipulation des données internes, la gestion des menus, des fenêtres, de l'écran, de la mémoire...

Elles sont regroupées par fonction dans des managers dont voici la liste :

- **Quickdraw** : dessins sur l'écran.
- **Menu manager** : gestion des menus.
- **Window manager** : gestion d'une fenêtre.
- **Text Editor** : édition de texte.
- **File manager** : gestion des fichiers.
- **Event manager** : gestion des événements.
- **Control manager** : gestion d'une fenêtre de dialogue.
- **Dialog manager** : gestion des dialogues.

Pour utiliser l'une des routines d'un manager, il faut :

- déclarer le manager pour une édition de liens ultérieure.
 - initialiser le manager (affectations de variables globales, réservation de place mémoire, etc...).
- On a alors directement accès aux constantes, variables, types prédéfinis, et aux routines.

* **Quickdraw.**

Sous le nom de *Quickdraw* sont regroupés environ 150 procédures et fonctions, qui supportent toute la gestion graphique. On peut dessiner :

- des caractères suivant différentes polices;
- des lignes de différente largeur et longueur;
- des formes géométriques : rectangles, cercles, ovales, polygones.

Pour créer des graphiques, il est nécessaire de définir quatre concepts de base : les *coordonnées planes*, le *point*, le *rectangle* et la *région* (pour leur définition Pascal, consulter la bibliographie annexe).

- les *coordonnées planes* sont des nombres entiers compris entre -32768 et 32767.
- les *points* : les coordonnées planes définissent 4 294 967 296 points individuels, chacun à l'intersection d'une ligne horizontale et verticale.
- le *rectangle* : Il est défini par deux points. Cette figure géométrique est d'une importance particulière puisque toute structure, aussi complexe soit elle, possèdera toujours parmi ses composantes le rectangle circonscrit qui définira la taille totale de la figure.
- la *région* : Quickdraw offre la possibilité de manipuler des ensembles de points répartis de manière totalement arbitraire sous la forme d'une seule structure nommée région.

Ces quatre concepts ne possèdent aucune réalité graphique, autrement dit ils ne peuvent pas être affichés à l'écran. Par opposition, il existe quatre types d'objets ayant une réalité graphique nommés *bitImage*, *bitMap*, *pattern*, *cursor*.

- le *bitImage* : c'est un ensemble de bits en mémoire. L'écran du Mac Intosh, lui-même, constitue un *bitImage* de 21 888 octets permettant d'afficher 175 014 pixels, chaque bit correspondant à un pixel. La taille de l'écran est de 342 pixels de haut par 512 pixels de large et la largeur de son *bitImage* est de 64 octets.

- le *bitMap* : en associant à un *bitImage* un système de coordonnées planes, on obtient la notion de *bitMap*, qui sera composé de trois parties : un pointeur sur un *bitImage*, la largeur de ce dernier en octets et le rectangle circonscrit qui procure au *bitMap* à la fois ses dimensions et un système de coordonnées.

- le *pattern* : c'est une image de 64 bits servant à définir un motif répétitif ou une couleur.

- le *cursor* : il est présent sur l'écran et associé au mouvement de la souris.

Quickdraw possède un type de variables nommé *grafport* qui définit où et comment auront lieu les opérations graphiques à venir. C'est la structure qui permet à un programme de construire des fenêtres. Il peut y en avoir plusieurs ouvertes simultanément et chacune a son propre système de coordonnées.

* Menu Manager.

Le gestionnaire des menus permet à l'utilisateur d'examiner les menus déroulants mis à sa disposition sans obligation d'en sélectionner un. Une exploration de quelques secondes de la barre de menus suffit donc à trouver la fonction recherchée.

* Window Manager.

Le gestionnaire des fenêtres permet de créer, d'activer, de déplacer et de changer la taille d'une fenêtre, la fenêtre étant la surface de travail de l'application. C'est un outil très puissant, il garde la trace des fenêtres empilées; on peut aussi manipuler ces fenêtres sans tenir compte de leur ordre d'empilement.

* Text Editor.

Le gestionnaire de texte s'occupe du traitement de texte.

* **File Manager.**

Le gestionnaire des fichiers est l'outil essentiel pour transmettre des informations de Mac Intosh vers un périphérique ou réciproquement. Les fichiers sont des séquences d'octets portant un nom et parfois regroupés dans des volumes (disques). Ce gestionnaire va permettre de lire ou d'écrire ces octets dans les fichiers.

* **Event Manager.**

Le gestionnaire des événements est le lien essentiel entre l'utilisateur et le programme. Lorsque l'utilisateur appuie sur le bouton de la souris, tape sur le clavier, insère un disque dans le lecteur, le programme en cours en est averti grâce à un événement.

* **Control Manager.**

Le gestionnaire de contrôles permet d'utiliser des contrôles. Un contrôle se définit comme un objet graphique avec lequel l'utilisateur, en se servant de la souris peut manipuler l'information ou modifier la manière dont celle-ci est présentée. Les barres de défilement vertical et horizontal en sont un exemple.

* **Dialog Manager.**

Le gestionnaire des dialogues permet de manipuler des fenêtres de dialogue. Un dialogue est une fenêtre qui apparaît sur l'écran lorsque l'application a besoin d'informations supplémentaires pour accomplir une tâche.

b) Les Ressources.

La notion de ressources permet la séparation du code et des données. L'intérêt des ressources est manifeste, elles permettent en effet de :

- Modifier les données sans recompiler le programme ;
- Réutiliser des ressources appartenant à d'autres applications ;
- Préserver la clarté et la lisibilité d'un source.

2.2) Choix du langage Fortran et Spécifications.

2.2.1) *Choix du langage Fortran.*

Le code d'éléments finis Modulef a été intégralement programmé en fortran, et c'est ce langage qui a été choisi pour l'application Treillis afin d'en assurer l'homogénéité, l'efficacité ainsi que la maintenance. D'autre part, il existe un compilateur fortran sur Mac Intosh qui permet grâce à des modules d'interface d'utiliser toutes les routines disponibles en R.O.M., ainsi on a pu mettre à profit toutes les ressources logicielles de ce micro.

2.2.2) *Spécifications.*

a) Les données à manipuler.

Les primitives correspondant au traitement des poutres doivent permettre de manipuler cinq ensembles de données, qui sont :

- *la géométrie* : les sommets (extrémités des poutres), les poutres repérées par deux numéros de sommets.

- *les données* : elles regroupent la section, le moment d'inertie, le module d'Young pour chaque poutre.

- *les charges ponctuelles* : composantes de la force et du moment s'exerçant en un point.

- *les charges réparties* : composantes de la force et du moment s'exerçant sur une poutre.

- *les limites* : composantes des déplacements et de la rotation imposées.

A chaque ensemble de données est associé une table :

* *La table des sommets* contient les numéros des sommets qui sont fixés à l'avance et les coordonnées à saisir de type réel.

* *La table des poutres* contient les numéros des poutres fixés par avance et les numéros des sommets à saisir.

* *La table des charges ponctuelles* contient un numéro de sommet, les forces et le moment qui lui sont appliqués.

* *La table des charges réparties* contient un numéro de poutre, les forces et le moment qui lui sont appliqués.

* *la table des données* contient un numéro de poutre et ses caractéristiques mécaniques.

* *la table des limites* contient un numéro de sommet, les déplacements et rotations associés.

b) Les fonctionnalités désirées.

Le menu principal doit offrir les possibilités suivantes :

- opérations sur les fichiers.
- opérations sur les tables.
- choix des objets à manipuler.
- choix de l'affichage des résultats.
- enchaînement sur la résolution.

Les primitives correspondantes au traitement graphique des poutres doivent permettre de :

- visualiser le dessin de la structure en dimension deux.
- créer ou supprimer un sommet, une poutre ou un noeud grâce à la souris.
- visualiser les points où sont imposés soit des conditions aux limites, soit des forces ou moment, ainsi que les poutres dont on a saisi les caractéristiques.

Dans la communication souris-écran-microprocesseur, on ne retiendra que deux commandes :

- le clic : un coup bref sur le bouton de la souris.
- le glisser de A jusqu'à B : on vise A, on appuie sur le bouton de la souris, on vise B en gardant le bouton appuyé, on lâche le bouton.

III. Description externe ou guide pour l'utilisateur.

Ce chapitre doit permettre à un utilisateur, peu familiarisé au maniement du Mac Intosh, de se servir de l'application Treillis pour résoudre un problème de calcul de structures de poutres en dimension deux, dans un cas statique. Il présentera tout d'abord le principal concept graphique : la fenêtre, puis les diverses fonctionnalités de l'application.

3.1) Généralités.

Avant de se lancer sur le clavier, quelques généralités sont nécessaires à la bonne compréhension de ce qui va suivre. Un écran type ,sur Mac Intosh, contient divers éléments dont les appellations sont fixées; le plus employé est la *fenêtre* .

Les fenêtres constituent un support capable de visualiser toute sorte d'informations. Elles possèdent un titre centré, des barres de défilement horizontal ou vertical, ainsi qu'un contrôle de taille dans leur coin inférieur droit. La fenêtre active , dont la barre de titre est contrastée, est celle située au premier plan . Il ne peut y avoir qu'une seule fenêtre active à un instant donné .

Il y a un certain nombre d'actions standards :

- *cliquer n'importe où*, à l'intérieur d'une fenêtre inactive, la rend active instantanément, en la positionnant au premier plan et en contrastant sa barre de titre.

- *cliquer dans la case* de fermeture d'une fenêtre active referme celle-ci : elle disparaît.

- *déplacer la souris* alors que son bouton a été appuyé dans la barre de titre déplace le contour de la fenêtre au travers de l'écran. Lorsque l'utilisateur relâche le bouton de la souris, la fenêtre est alors redessinée à sa nouvelle position.

- *déplacer la souris* alors que son bouton a été appuyé dans les contrôles de taille modifie la taille de la fenêtre.

3.2) Le menu principal et les diverses fonctionnalités.

FICHIER TABLES GEOMETRIE RESOLUTION

Cinq fonctions peuvent être sélectionnées :

- *La pomme* : elle permet l'utilisation des outils standards de bureau tels que l'album, l'horloge le calepin, la calculatrice...

- *La fonction Fichier* : elle assure la gestion des fichiers créés par l'application.

- *La fonction Tables* : elle permet de saisir les données du problème traité.

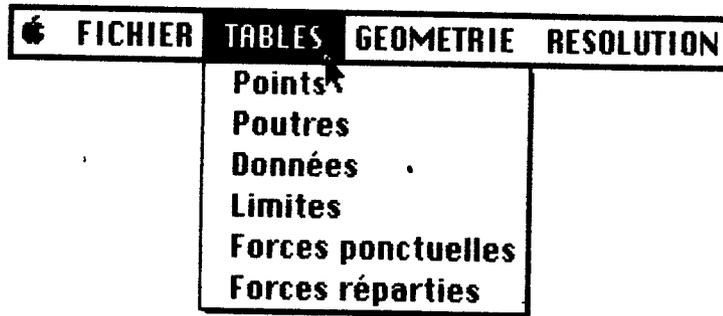
- *La fonction Géométrie* : elle offre la possibilité de saisir graphiquement un treillis bidimensionnel de poutres.

- *La fonction Résolution* : elle permet la saisie du code à utiliser, de la méthode, ainsi que des résultats que l'utilisateur doit connaître. Ces options ont été prévues principalement pour un code autre que Modulef. En fait cette fonction servira principalement à lancer l'exécution du programme qui génère les structures de données nécessaires à Modulef pour la résolution.

3.3) L'aspect saisie clavier : le menu "tables".

Il gère la saisie des données réparties suivant 6 tables qui contiennent :

- les coordonnées des points.
- les numéros des points extrémités de chaque poutre.
- les conditions aux limites imposées en un point.
- les composantes de la force ponctuelle et du moment appliquées en un point.
- les composantes de la force répartie et du moment appliquées sur une poutre.



- **la table des points** : l'utilisateur doit donner pour chaque point prénommé, ses deux composantes réelles dans le repère de son choix. Il peut se déplacer dans la table, soit de case en case en appuyant sur la touche "retour charriot", soit en cliquant dans la case désirée.

Il dispose, de plus, de la barre de défilement pour parcourir la table, du carré en haut à gauche pour la fermer, de la barre de titre pour la déplacer. Lorsqu'il saisit des données un contrôle direct est réalisé : seuls les caractères "+", "-", ".", "E" et les chiffres sont pris en compte.

Point	Abscisse	ordonnee
7	-3.E22	12.54376567
8	-12309867	1.2E16
9	2.345E-4	
10		
11		12
12		
13		

Inserer Supprimer

Il existe deux fonctions d'aide à l'agencement des données :

- *Inserer* : pour l'activer, il faut cliquer dans la case contenant le libellé, elle apparaît alors en inverse vidéo. A partir de ce moment, chaque fois que l'utilisateur cliquera une case de la colonne point, il insérera à cet endroit une nouvelle ligne dans la table (la numérotation restera toujours contigüe) . Afin d'assurer une prénnumérotation cohérente, chaque insertion est répercutée sur les autres tables. On renumérote ainsi dans toutes les tables les points ayant changés de numéros lors de l'insertion.

L'utilisateur peut aussi cliquer une case de la colonne abscisse ou ordonnée, il insère alors une case uniquement dans cette colonne. *Dans ce cas il n'y a aucune répercussion sur les autres tables* car les mouvements des données deviennent trop coûteux en temps à gérer.

Naturellement aucune insertion n'est possible après le dernier point de la table. Pour désactiver la fonction insérer il suffira à l'utilisateur de recliquer dans la case "insérer" qui ne sera alors plus en inverse vidéo.

- *supprimer* : pour activer ou désactiver la fonction supprimer, on procède de la même façon que pour insérer. En cliquant dans une case de la colonne point, l'utilisateur supprimera la ligne correspondante. La numérotation restant toujours contigüe, une répercussion sur les autres tables est également assurée, le point supprimé n'existe plus dans les autres tables.

L'utilisateur peut aussi cliquer une case de la colonne abscisse ou ordonnée, il supprime alors une case uniquement dans cette colonne. *Dans ce cas il n'y a aucune répercussion sur les autres tables.*

Naturellement aucune suppression n'est possible après le dernier point de la table.



L'utilisateur peut rentrer ses données dans la table de manière anarchique. Cependant lorsqu'il désactive sa table, celle-ci est compactée : on supprime les lignes vides ou incomplètes en mettant à jour les autres tables.

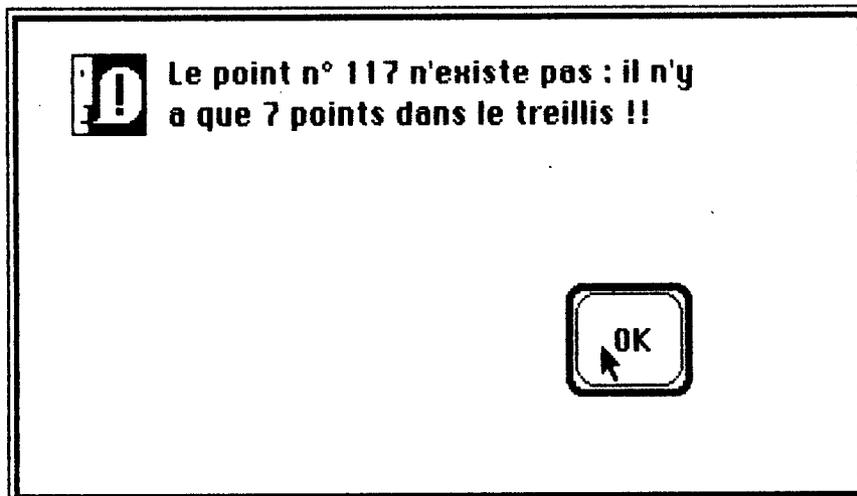
- **la table des poutres** : l'utilisateur doit donner pour chaque poutre prénommée, ses deux points extrémités saisis auparavant dans la table des points.

table des poutres			
Poutre	Point1	Point2	
1	1	2	
2	117		
3	2	3	
4	3	4	
5	4	5	
6			
7			
Insérer		Supprimer	

L'utilisateur dispose des mêmes facilités que pour la table des points : de la barre de défilement et des deux fonctions *supprimer* et *insérer* .



Si l'utilisateur essaye de rentrer dans la table des poutres, un numéro de point n'ayant pas été déjà défini, il verra apparaître à l'écran une fenêtre d'alerte lui indiquant la cause de son erreur.



- **la table des données** : l'utilisateur doit donner pour chaque poutre prénúmerotée, ses caractéristiques physiques : E son module d'Young, I le moment d'inertie de sa section droite, S sa section droite. Pour plus de souplesse, il pourra sauvegarder sur fichier une saisie incomplète. Cette table, contrairement aux précédentes ne sera pas compactée, lorsqu'elle est désactivée.

table des donnees			
Poutre	E	I	S
1	70000	50000	121
2		20000	
3	30000		50000
4			
5	30000	30000	50000
6			
7			
Insérer		Supprimer	Repeter

L'utilisateur dispose des mêmes facilités que pour la table des points : de la barre de défilement et des deux fonctions *supprimer* et *insérer* . Il peut utiliser la fonction supplémentaire :

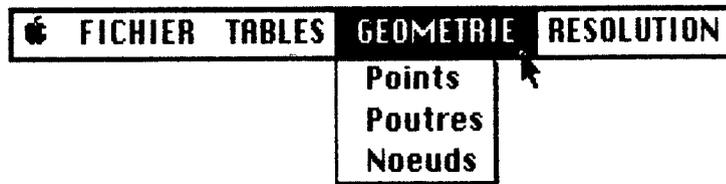
- *répéter* : après l'avoir activé , chaque case ou ligne non vide cliquée sera mémorisée et dupliquée en cliquant sur une case ou une ligne vide. Ce traitement s'effectuera indépendamment du déplacement dans la table à l'aide de la barre de défilement.



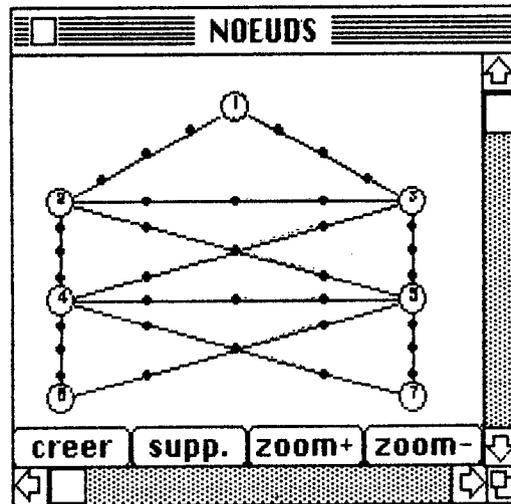
Si l'utilisateur essaye de rentrer dans la table des données, un numéro de poutre n'ayant pas été déjà définie, il verra apparaître à l'écran une fenêtre d'alerte lui indiquant la cause de son erreur.

3.4) L'aspect saisie graphique : le menu géométrie.

Il permet de visualiser ou de saisir dans une fenêtre la géométrie d'une structure bidimensionnelle de poutres. Il propose trois options points ,poutres ou noeuds que l'on sélectionne suivant la nature de ce que l'on veut saisir graphiquement.



Lors de la sélection de "points" , de "poutres" ou de "noeuds" la fenêtre graphique ci-dessous est activée, son titre rappelle à l'utilisateur l'option choisie.



On dispose ici des deux barres de défilement, du contrôle de taille et des quatre fonctions que l'on active en cliquant dans leur libellé :

- **Créer** : la nature de l'objet à créer est désignée par le titre de la fenêtre. Il suffit de cliquer dans la case de libellé "créer" pour activer celle-ci.

Pour créer un point, il faut cliquer sur la position souhaitée. Le point aura le dernier numéro libre de la table des points et ses composantes y seront inscrites après avoir été calculées en fonction de l'échelle courante.

Pour créer une poutre, il est nécessaire de reconnaître ses deux points extrémités en cliquant successivement chacun d'eux. L'utilisateur s'il le veut pourra cliquer un point, faire déplacer sa structure à l'aide des barres de défilement et cliquer un autre point. La jonction s'effectuera même si les deux points ne sont pas simultanément apparents. La poutre aura le dernier numéro libre de la table des poutres et les numéros de ses extrémités y seront inscrites.

Pour créer des noeuds sur une poutre il faut tout d'abord saisir, lors de l'activation de la fonction créer, dans une fenêtre de dialogue :

- le nombre de noeuds à créer.
- la distribution des noeuds :
 - équidistants.
 - en progression géométrique : il faudra alors saisir la raison de cette progression.
- si tout est à mailler.

Si tout est à mailler, le maillage s'effectuera automatiquement. Dans le cas contraire, il suffira de cliquer sur une poutre (même si elle n'est que partiellement apparente) pour la segmenter suivant les caractéristiques préalablement saisies.

- **Supprimer** : La nature de l'objet à supprimer est indiquée par le titre de la fenêtre. Après avoir sélectionné la fonction "supprimer", il suffit de cliquer sur un objet pour le voir disparaître. Cette suppression graphique s'accompagne des différentes mises à jour nécessaires dans les tables concernées.

- **Zoom+** : Cette fonction permet d'agrandir la taille de la géométrie dessinée dans la fenêtre. Cet agrandissement restera uniquement graphique et ne sera pas répercuté dans les tables. Pour assurer la cohérence avec le repère de l'utilisateur, une échelle-translation est calculée à chaque modification de la taille.

- **Zoom-** : Cette fonction permet de diminuer la taille de la géométrie dessinée dans la fenêtre. Cette diminution restera uniquement graphique et n'aura pas de conséquence sur les tables.

- le sous-menu **"points"** : En le sélectionnant, la fenêtre graphique est activée avec le titre "points". Les objets manipulés seront alors des points.

- le sous-menu **"poutres"** : En le sélectionnant, la fenêtre graphique est activée avec le titre "poutres". Les objets manipulés seront alors des poutres.

- le sous-menu **"noeuds"** : En le sélectionnant, la fenêtre graphique est activée avec le titre "noeuds". Les objets manipulés seront alors des noeuds.

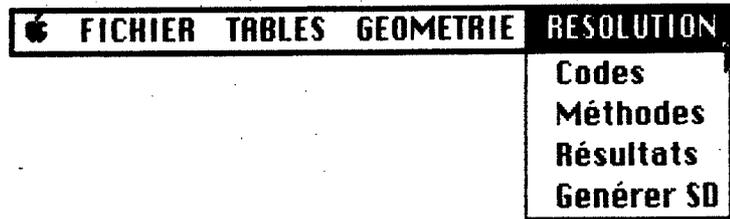
Remarques :

- Les points et les poutres sont apparents dans les trois modes (points, poutres et noeuds), afin de ne pas ralentir l'exécution les noeuds ne sont visualisés qu' en mode noeuds.

- On peut effacer un maillage en sélectionnant la fonction créer avec 0 comme nombre de noeuds et l'option "tout mailler".

3.5) La fonction résolution.

Elle permet à l'utilisateur de choisir le *code* sous lequel sera résolu son problème, la *méthode* à employer et les *résultats* à rendre. Elle permet d'autre part de lancer le programme qui va *générer* les structures de données Modulef pour un treillis dont on a saisi toutes les caractéristiques.



- La fonction "codes" : Cette fonction permet à l'utilisateur d'indiquer le code d'éléments finis qui traitera son problème, et celui qui exploitera ses résultats. Il lui suffit de cliquer dans les deux options de son choix.

Cette fonction n'aura d'utilité que si l'application treillis sert de préprocesseur à un autre code que Modulef.

EXECUTION SOUS

CALCULS : ils seront faits sur le code suivant

ISI MEF MODULEF NASTRAN

RESULTATS : ils seront exploités sur le code suivant

ISI MEF MODULEF NASTRAN

- La fonction "méthodes" : Elle propose à l'utilisateur, dans le cas où son problème est statique, une résolution par la méthode de Cholesky ou du gradient conjugué. L'utilisateur clique la méthode choisie, celle-ci ne sera prise en compte que pour un code d'éléments finis autre que Modulef.

RESOLUTION DU SYSTEME LINEAIRE

PROBLEME STATIQUE

METHODE DE CHOLESKI

METHODE DU GRADIENT CONJUGUE

- La fonction "résultats" : Elle permet à l'utilisateur de sélectionner les résultats qu'il désire connaître, suivant la nature de son problème : statique ou dynamique. Il sélectionne ce qu'il désire en cliquant les cases de son choix. L'application ne se limitera dans un premier temps qu'au traitement des cas statiques et au calcul des déplacements.

SORTIE DES RESULTATS

STATIQUE

Deplacements aux noeuds

Contraintes dans les elements

Reactions aux appuis

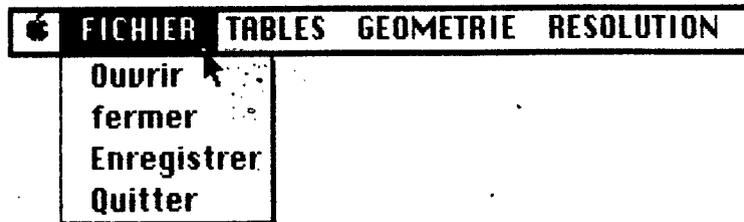
DYNAMIQUE

Frequences propres

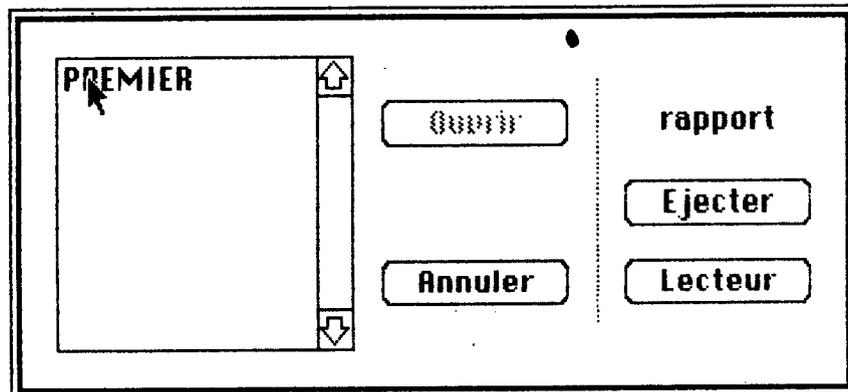
Deformees propres

- La fonction "générer 3D" : Elle permet de lancer l'exécution du programme qui crée les structures de données nécessaires à Modulf pour la résolution, puis de résoudre le problème.

3.6) La fonction fichier : Cette fonction propose quatre options assurant la gestion des fichiers.



- La fonction "ouvrir" : Cette fonction propose à l'utilisateur de charger ses données à partir d'un fichier . Elle appelle elle-même la fonction sauvegarder ; l'utilisateur a ainsi la possibilité de sauvegarder une saisie antérieure . Il sélectionne le fichier désiré en cliquant deux fois son nom .



- La fonction "fermer" : Cette fonction permet à l'utilisateur de fermer les tables en sauvegardant leur contenu dans un fichier , puis de recommencer une saisie pour un nouveau problème . Le contenu des tables est réinitialisé.

- **Le fonction "enregistrer"** : Cette fonction permet à l'utilisateur de sauvegarder ses données dans un fichier .Il lui suffit d'entrer le nom du fichier souhaité. Le nom du dernier fichier ouvert apparaît par défaut .

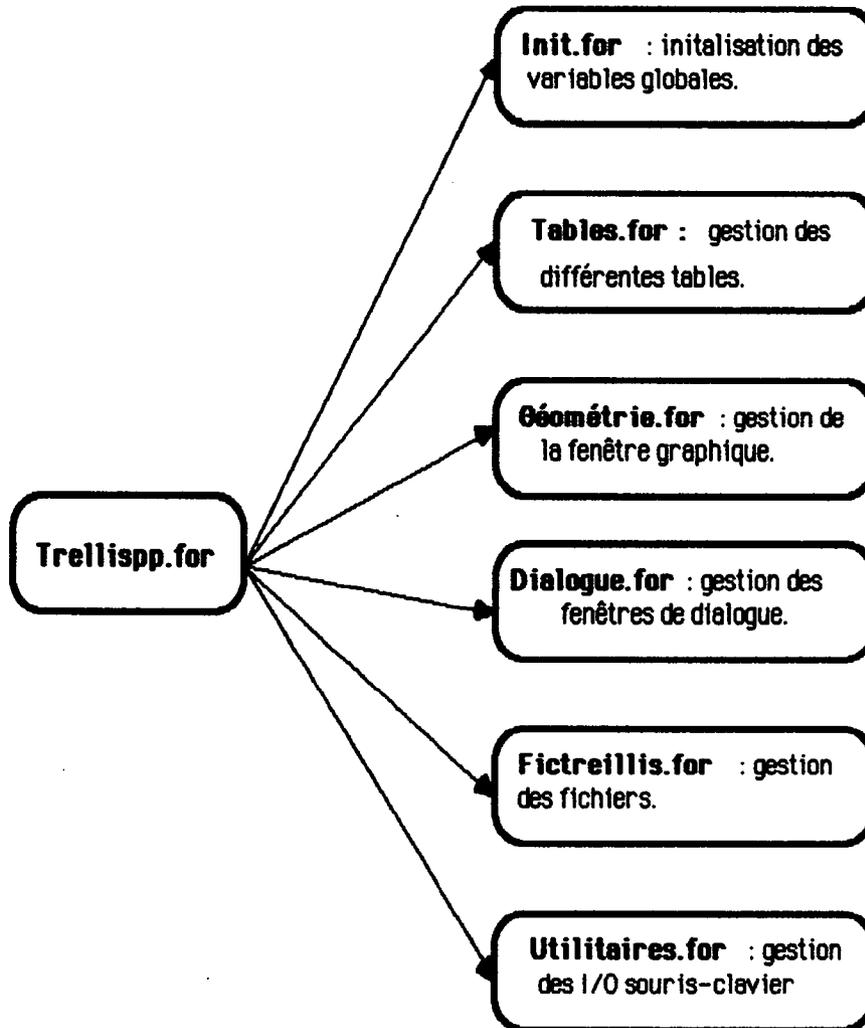
Sauver le tableau dans :		rapport
<input type="text" value="TREILLIS"/>		Ejecter
Enregistrer	Annuler	Lecteur

- **Le fonction "quitter"** : Cette fonction permet à l'utilisateur de quitter l'application. Par sécurité, elle appelle la fonction **enregistrer** .

IV. Description interne ou guide pour le développeur.

Ce chapitre est destiné à faciliter la maintenance et le développement de l'application TREILLIS. Il présentera l'architecture modulaire du logiciel, les structures de données employées, la structure des fichiers et enfin un algorithme général du programme.

4.1) Architecture des différents modules de l'application.



4.2) Les Contraintes de programmation sous Mac Intosh.

La structure du programme et le nombre impressionnant de variables globales sont dues à une contrainte dans la gestion des événements, contrainte à laquelle on ne peut pas échapper.

Tous les événements (souris, clavier, activation, désactivation ou remise à jour d'une fenêtre...) sont gérés dans une boucle du programme principal, y compris les événements intervenant dans les procédures qui gèrent les options du menu et qui peuvent être situées très bas dans la hiérarchie de l'application.

Les événements sont donc tous gérés dans le programme principal, ce qui implique que le programme est en attente au niveau le plus haut.

On a donc comme principal inconvénient l'*obligation de définir des variables globales* qui ne servent souvent qu'à l'intérieur d'une seule procédure : en effet, si l'on veut, entre deux événements, mémoriser l'état d'une variable, il faut que celle-ci soit globale, car on quitte systématiquement la procédure pour remonter au programme principal.

Cela explique la floraison contre nature des variables globales dans notre programme.

4.3) La structure des données.

- **les tables** : elles sont au nombre de six et chacune d'elle est identifiée par une constante (le nombre maximal prévu *Nbtab_max* est 6).

Ces constantes sont :

- *Tab_points* : identificateur de la table des points égale 1.
- *Tab_poutres* : identificateur de la table des poutres égale 2.
- *Tab_données* : identificateur de la table des données égale 3.
- *Tab_limites* : identificateur de la table des limites égale 4.
- *Tab_forces* : identificateur de la table des forces ponctuelles égale 5.
- *Tab_densités* : identificateur de la table des forces réparties égale 6.

Elles se composent de colonnes qui servent soit à la saisie d'entiers, soit à la saisie de réels (le maximum de colonnes *Nbcol_max* est 4, le maximum d'éléments *nb_points* est 60). Pour rester le plus générique possible et permettre la création de nouvelles tables sans remettre en cause leur gestion, la structure de donnée adoptée est représentée par une suite de colonnes typées. On aura donc pour chacune d'elle les renseignements suivants :

(Dans tout ce qui suit on aura $1 \leq la_table \leq Nbtab_max$
 $0 \leq la_colonne \leq Nbcol_max-1$
 $1 \leq l_element \leq nb_points$)

- Un tableau de handle (de type integer) *mywindow* (*la_table*) désignant la fenêtre associée à chaque table (mis à 0 si la table n'est pas présente à l'écran). Un handle est un pointeur de pointeur sur un bloc situé dans le "heap" de la mémoire (on rappelle que le heap est un espace mémoire où sont gérées les variables dynamiques). Le type handle est un concept propre au Mac Intosh. Il permet au système de gérer plus facilement la zone heap de la mémoire.

- Le numéro de sa première et de sa dernière colonne : *nbcol_pre* (*la_table*) et *nbcol_der* (*la_table*).

- Le *type* de chacune de ses colonnes. Il se trouve dans le tableau de caractères: *Typ* (*la_colonne*,*la_table*); pour chaque table et chaque colonne, les différents types sont représentés dans le tableau ci-dessous par :

- "i" : type entier.
- "r" : type réel.
- "a" : colonne inexistante

la table \ la colonne	0	1	2	3
Tab_points	i	r	r	a
Tab_poutres	i	i	i	a
Tab_données	i	r	r	r
Tab_limites	i	r	r	r
Tab_forces	i	r	r	r
Tab_densités	i	r	r	r

- Le tableau de correspondance d'integer*2

C (la_colonne, la_table), ci-dessous. Il donne le numéro de la colonne dans le tableau où sont stockés les éléments de son type. Il y a deux tableaux :

Col_int (L'élément, C(la_colonne, la_table)), tableau des colonnes d'entiers (integer*2) et *Col_real* (L'élément, C(la_colonne, la_table)), tableau de colonnes des réels (type real).

Les indices soulignés indiquent le numéro d'une colonne dans le tableau d'entier *Col_int*.

Les autres indiquent le numéro d'une colonne dans le tableau d'entier *Col_real*.

la colonne la table	0	1	2	3
Tab_points	<u>1</u>	1	2	
Tab_poutres	2	<u>3</u>	4	
Tab_données	<u>5</u>	3	4	5
Tab_limites	<u>6</u>	6	7	8
Tab_forces	7	9	10	11
Tab_densités	<u>8</u>	12	13	14

- Le tableau de booléen (type boolean*1)

Set_real (L'élément, C(la_colonne, la_table)). Il permet de savoir si "L'élément" a été saisi dans la colonne. Pour le tableau *Col_int*, les entiers étant toujours positifs, la valeur initiale (*pt_init* = -1) a pu être affectée; elle sert de valeur de comparaison pour déterminer l'existence de "L'élément".

- Le tableau de handles (integer) *col_hdlr* (la_table). Il permet d'accéder à chaque buffer de texte, nécessaire à toute saisie et associé à "la_table".

- Le tableau d'integer*2 *pre_lignvis* (la_table). Il contient le numéro de la première ligne apparente dans "la_table".

- Le tableau d'integer*2 *num_der* (la_table). Il donne le nombre de lignes saisies dans "la_table".

- Le tableau d'integer*2 *indcase* (la_table). Il précise la position

relative du curseur dans "la_table".

- Le tableau d'integer *control* (la_table). C' est un tableau contenant des pointeurs sur les contrôles associés à la table.

- Le tableau d'integer*2 *rgeom* (4,la_ligne,la_table). C' est un tableau qui définit les coordonnées des rectangles associés aux cases de "la_table". Ils sont destinés à servir de support graphique au curseur.

- **le graphique** : Les instructions graphiques permettant d'afficher la géométrie sont sauvegardées dans une variable (picture) de type integer.

4.4) La structure des fichiers.

Les fichiers sous Mac Intosh sont des séquences d'octets portant un nom ; l'écriture ou la lecture de ces octets est assurée par le gestionnaire des fichiers . Les fichiers sont caractérisés par un nom et un certain nombre d'autres paramètres :

- Le type du fichier codé sur quatre caractères ASCII : (APPL) application, fichier text (TEXT)...

- La signature du programme créateur qui permettra à l'utilisateur lorsqu'il ouvre un document, de lancer l'exécution du programme qui l'a créée.

Un fichier est défini par son premier octet (octet numéro zéro) ; sa fin logique se trouve un octet après le dernier octet du fichier, et sa fin physique est située un octet après le dernier bloc alloué.

Nous avons organisé la structure de nos fichiers dans l'ordre séquentiel suivant :

2 octets

nombre de points
nombre de poutres
nombre de données
nombre de limites
nombre de forces ponctuelles
nombre de forces réparties

12 octets

Pour les tables des *points* et des *poutres*, deux données par élément sont stockées.

abscisse, ordonnée de chaque point : 2 réel, **8 octets**

numéros des points extrémités de chaque poutre : 2 entiers, **8 octets**

La table des données n'est pas compactée. Pour *chaque poutre* dont l'utilisateur a saisi la section, on sauvegarde :

- le numéro de la poutre : 1 entier, **2 octets** ;

Pour le Module d'Young E , l'inertie I et la section S de la poutre on a :

- un booléen qui sert à déterminer l'existence de la donnée, **1 octet** ;

- la valeur de la donnée, **4 octets** .

Pour les tables des *limites*, *forces ponctuelles* et *forces réparties* on a :

Pour les trois données de chaque élément :

- un booléen qui sert à déterminer l'existence de la donnée, **1 octet** ;

- la valeur de la donnée, **4 octets** .

Pour chaque *poutre* :

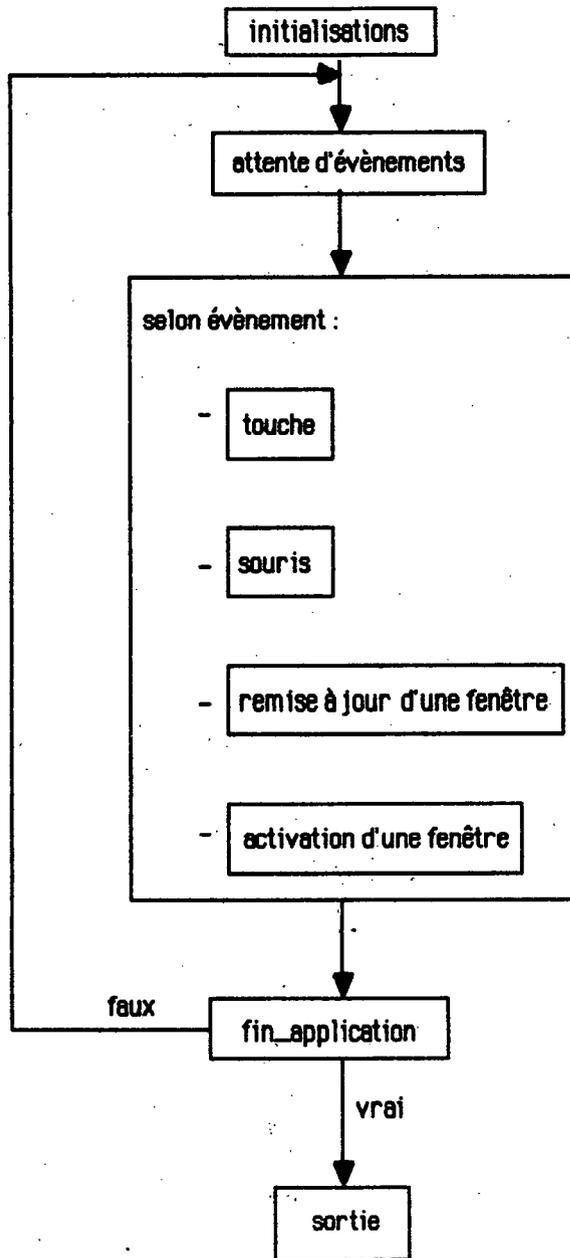
- son nombre de noeuds : 1 entier , *1 octet* ;
- la raison géométrique : 1 réel , *4 octets*.

pour chaque *boutton* des fenêtres de dialogue :

- son code : 1 entier , *4 octets*.

4.5) L'algorithme du programme principal.

Vu la longueur du programme , il serait long et fastidieux de décrire les différents algorithmes. Nous avons choisi de ne présenter que l'algorithme du programme principal afin de montrer simplement la philosophie de programmation sur Mac Intosh.



V. Conclusion.

L'application TREILLIS s'insère dans le cadre spécifique de l'élaboration d'un préprocesseur ergonomique pour le traitement numérique, sur Mac Intosh, d'un problème bidimensionnel de poutres par éléments finis. Le logiciel réalisé se présente en effet comme un produit fini :

- Il permet une saisie de toutes les données nécessaires au traitement par éléments finis d'un problème de poutres (points, poutres, forces, conditions aux limites...).
- Il offre des commodités de saisie dans les tables (fonctions insérer, supprimer, répéter).
- Il permet une visualisation graphique de la saisie de la géométrie de la structure à étudier .
- L'utilisateur peut sauver et récupérer ses données dans des fichiers .
- L'utilisateur peut enchaîner l'exécution du programme générant les structures de données propres à Modulf et la résolution de son problème.

Son avantage est qu'il permet de résoudre un problème statique de poutres en saisissant de manière agréable le strict minimum des données. Bien sûr, d'autres utilitaires pourraient être ajoutés tel que la visualisation des forces appliquées , une saisie graphique des conditions au limites , une segmentation automatique d'une poutre

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique