

ETAT DE L'ART DANS LE DOMAINE DE L'INFOGRAPHIE INTERACTIVE

M. LUCAS

Laboratoire IMAG
BP 53
38041 GRENOBLE-Cedex (France)

RESUME

Nous présentons dans cet article les principaux acquis réalisés ces dernières années dans le domaine de l'infographie interactive. Les principaux sujets abordés concernent le matériel (nouveaux écrans, utilisation de microprocesseurs), le logiciel (les algorithmes de production d'images réalistes, les efforts de normalisation) et les applications les plus classiques (animation, illustration de programmes, conception assistée par ordinateur).

INTRODUCTION

Nous assistons depuis quelques années à une véritable explosion du marché des consoles de visualisation. Le nombre d'appareils utilisés croît de façon exponentielle, de nouveaux modèles sont proposés chaque année par des constructeurs de plus en plus nombreux, le champ des applications utilisant les techniques de communication graphiques interactives s'élargit sans cesse. Ce succès est dû aux progrès réalisés tant du point de vue matériel que du point de vue logiciel, progrès permettant d'offrir aux utilisateurs des matériels à des prix de plus en plus faibles et des algorithmes de plus en plus perfectionnés. Nous nous proposons de mettre en valeur les principaux acquis de ces précédentes années, en explorant tour à tour le matériel, le logiciel et les applications.

1. LE MATERIEL

Une console de visualisation se compose essentiellement d'une surface de visualisation (écran), d'un processeur graphique et de dispositifs de communication. Nous étudierons séparément ces trois composants.

Les surfaces de visualisation

On distingue deux types de surfaces de visualisation :

- les surfaces de dessin au trait, où le composant élémentaire est le point ou le segment de droite, et qui permettent de produire des représentations schématiques désignées sous le nom de dessins,
- les surfaces à pointillage, où le composant de base est un carreau élémentaire ("pixel") auquel on associe une couleur, et qui permettent de produire des images, assemblages de taches colorées.

Les surfaces de visualisation au trait étaient les plus répandues, et ces dernières années ont permis de constater quelques progrès à leur

niveau, concernant essentiellement le nombre d'éléments affichables (dans le cas des surfaces à régénération), la taille de l'écran (passant de 30 x 30 cm à 40 x 40 et même 70 x 100 cm), la possibilité d'utiliser des couleurs, (trois ou quatre en général, certains équipements permettant d'atteindre une dizaine de couleurs), la définition (4096 x 4096 points sur un écran de 38 x 38 cm). Parmi les dispositifs les plus spectaculaires, citons une console de visualisation à rayon laser, qui permet d'obtenir des dessins d'une très grande précision directement sur microfilm, à une vitesse très élevée (10 mètres de tracé à la seconde).

Cependant, la plus grande révolution vient de l'apparition des surfaces de dessin à pointillage. Il s'agit essentiellement d'écrans de télévision auxquels on adjoint une mémoire d'entretien permettant de défluir la couleur et la luminosité de chaque point. Ces écrans autorisent la composition d'images utilisant jusqu'à 256 couleurs différentes pour les appareils ordinaires, que la couleur soit "vraie" (c'est-à-dire obtenue à partir du codage des trois fondamentales) ou "fausse" (c'est-à-dire obtenue à partir d'une table de codage liée à l'application). A l'heure actuelle, ces écrans ont le défaut de ne pas disposer d'une définition suffisante (en général 512 x 512 points), ce qui ne leur permet pas de supplanter les autres types de consoles. Cependant, de nouveaux matériels sont annoncés avec des définitions de 1024 x 1024 points, qui seront alors directement concurrentiels de certains écrans à balayage cavalier.

Les processeurs graphiques

Le processeur graphique est un petit ordinateur chargé de gérer les opérations d'entrée/sortie qui se déroulent au niveau de la console et de produire l'image ou le dessin sur l'écran. La caractéristique essentielle des progrès réalisés ces dernières années est l'emploi de plus en plus systématique de microprogrammes ou de microprocesseurs.

En ce qui concerne l'usage de microprogrammes, ils sont destinés en général à faciliter la gestion de la mémoire d'entretien des consoles à balayage cavalier. On trouve ainsi des microprogrammes permettant de mettre à jour les informations dans la mémoire, de les structurer et de délivrer des renseignements (valeurs de corrélation, positions etc) à leur sujet lors des opérations de dialogue.

En ce qui concerne l'usage des microprocesseurs, la modicité du coût de ceux-ci et leur banalisation a conduit à une prolifération anarchique de leur usage. En effet, les micro-processeurs sont utilisés aussi bien pour accélérer certains algorithmes de base (gestion de l'écran, génération de symboles, algorithmes élémentaires de visualisation) que pour permettre la mise en oeuvre de consoles évoluées "à la carte", très adaptées à une application donnée. Si l'on regarde de près les recherches qui se font dans ce domaine, on ne peut qu'être frappé de la multiplicité des solutions techniques qui sont retenues, de la diversité des fonctions qui sont mises en oeuvre, du peu de cohérence des choix qui sont faits quant aux primitives élémentaires à réaliser. La conclusion qui s'impose est double :

- l'avenir est bien évidemment à l'utilisation de ces techniques, ne serait-ce que parce que le coût du matériel ne cesse de décroître alors que le coût du logiciel ne cesse de croître,

- la maîtrise de la structure des logiciels graphiques interactifs n'est pas atteinte, ce qui est encore plus mis en valeur par les travaux de normalisation que nous évoquerons plus loin.

Les dispositifs de communication

Force est de constater ici une relative stagnation. Autant une grande richesse d'invention apparaît au niveau des écrans et des processeurs graphiques, autant une faible originalité se fait jour au niveau des dispositifs de dialogue. Ce sont les mêmes qui existent depuis l'apparition des premières consoles graphiques (photostyle, clavier de fonctions, clavier alphanumérique, tablette, manche à balai, etc ...), avec quelques petites améliorations. Le seul pas significatif concerne la mise à disposition de tablettes graphiques de grandes dimensions, offrant parfois plusieurs stylos releveurs, permettant ainsi de digitaliser des dessins de grandes dimensions (cartographie, par exemple) ou de traiter des objets tridimensionnels à partir de plusieurs vues, le même point étant repéré simultanément sur chaque vue par un stylo. Quelques dispositifs permettant de désigner des objets à l'aide du doigt ont vu le jour, mais pour l'instant leur avenir ne paraît pas très grand.

2. LE LOGICIEL

Nous évoquerons deux axes de travail :

- les recherches concernant les algorithmes de présentation d'images réalistes,
- les travaux de normalisation des logiciels graphiques.

La génération d'images réalistes

Cet axe de recherche est intimement lié à l'apparition des terminaux à pointillage. En effet, les chercheurs ont eu dès lors la possibilité de créer des images, et ont donc voulu engendrer des représentations du monde réel tel qu'il est perçu par l'oeil. Deux problèmes se posent alors : comment présenter un objet tridimensionnel sur une surface plane, et comment habiller cette représentation plane pour donner l'illusion de la réalité.

Les techniques de présentation d'objets tridimensionnels ont fait l'objet de travaux très poussés depuis dix ans. Les principaux problèmes traités concernaient l'élimination des parties de la scène à représenter qui sont invisibles pour un observateur donné. Il est tout à fait intéressant de constater que si la conception d'un algorithme d'élimination de parties cachées n'est plus un problème, on ne connaît pas à l'heure actuelle de critères permettant de déterminer si "le meilleur" algorithme existe ou est encore à trouver. Ceci se manifeste par un grand nombre de publications présentant de nouveaux algorithmes, qui ne sont parfois que des améliorations d'algorithmes bien connus. Chaque nouvelle publication annonce la découver-

te du meilleur algorithme qui soit, avec des chiffres extraordinaires à l'appui, mais les jeux d'essais excellents pour un algorithme se révèlent désastreux pour un autre. On peut donc conclure que la présentation d'objets avec élimination de parties cachées est devenue une chose banale et courante, mais que beaucoup de travail reste à faire pour parvenir à une parfaite connaissance de ces problèmes. En particulier, notons que peu d'algorithmes sont capables de traiter autre chose que des ensembles de polygones. De nombreux travaux sont en cours qui permettront de traiter des surfaces analytiques telles que les surfaces de COONS ou de BEZIER.

En ce qui concerne l'habillage des silhouettes obtenues par élimination des parties cachées, les chercheurs ont relativement bien maîtrisé les effets d'ombrage, de reflets et de transparence, permettant ainsi d'obtenir des images tout à fait extraordinaires représentant des objets de la vie courante comme on peut les voir sur une nature morte peinte à la main. Les travaux actuels portent sur la notion de texture, c'est-à-dire l'aspect extérieur de la surface d'un objet réel pour l'oeil. Un problème très important est par exemple celui de la représentation de paysages tels qu'ils sont vus d'avion, afin de construire des simulateurs de vol pour l'entraînement des pilotes. Un champ n'a pas une couleur unie, mais de subtiles variations de couleur perceptibles par l'oeil permettent d'identifier à coup sûr le type de culture (par exemple). Ce sont ces variations que l'on cherche à reconstituer lors du coloriage d'une surface.

En conclusion, notons deux choses :

- si des images extraordinaires ont été obtenues, elles résultent le plus souvent de "trucs", et il n'y a pas encore de théorie permettant de considérer ces générations comme un calcul simple et banal. Il est clair qu'un rapprochement avec les chercheurs qui font du traitement d'images ou de la reconnaissance de formes se fera sous peu.

- les temps de calcul pour obtenir ces images sont de l'ordre de la trentaine de minutes aujourd'hui. Cependant, si l'on se souvient que les algorithmes d'élimination de parties cachées demandaient autant de temps de calcul il y a quelques années et qu'aujourd'hui ils travaillent en temps réel, nous pouvons penser que dans cinq ans (les progrès sur le plan du matériel aidant) la génération d'images synthétiques permettra de réaliser des dessins animés en temps réel.

La normalisation des logiciels graphiques

L'apparente maîtrise des techniques graphiques et le fort développement du nombre d'utilisateurs a conduit très vite les chercheurs et les utilisateurs à se préoccuper des possibilités d'échange de programmes graphiques. Le mouvement qui a conduit à de nombreux travaux sur la normalisation du logiciel a pris naissance sur une initiative du groupe de travail IFIP WG 5.2 (Computer Aided Design). Après quelques réunions préparatoires, un séminaire de quatre jours eut lieu à SEILLAC (France), au mois de Mai 1976. Les personnes assistant à ce séminaire étaient d'origine très diverses, tant du point de vue des pays représentés (Amérique du Nord, Europe, Scandinavie) que du point

de vue professionnel (universitaires, utilisateurs, concepteurs, etc ...). Quatre résolutions furent adoptées :

- il est nécessaire de faire une claire distinction entre le logiciel graphique de base ("core") et le logiciel de description ("model - ling"),
- il est très souhaitable d'établir une taxonomie de la structure des programmes graphiques,
- il est nécessaire de donner une définition formelle des principaux concepts qui sont à la base des techniques graphiques,
- il est recommandé que les utilisateurs, concepteurs et réalisateurs de logiciels graphiques étudient en commun les logiciels distribués actuellement, afin de voir quelles modifications pourraient être apportées à ces produits, de manière à les rapprocher des recommandations émises par le séminaire.

A la suite de ce séminaire, de nombreux groupes de travail se sont formés dans tous les pays, s'attachant à mieux comprendre les problèmes posés par une éventuelle normalisation. Les grandes associations nationales et internationales se sont mises de la partie, ce qui fait que les résultats ne se sont pas faits attendre.

Les travaux les plus avancés à l'heure actuelle sont ceux du Graphics Standards Planning Committee, groupe de travail américain lié à l'ACM. Les membres de ce groupe ont publié en Juillet 1977 une proposition pour un logiciel graphique de base, proposition qui sert de base de travail aux associations de normalisation. Cette proposition est à l'heure actuelle discutée et amendée, tandis que de nombreuses réalisations sur des matériels différents, voient le jour et que la conformité des logiciels graphiques aujourd'hui distribués est vérifiée. Il s'agit donc d'un mouvement très important, qui a toutes les chances d'aboutir à une norme dans quelques années. Cependant, des critiques de fond ont été émises à l'égard de cette proposition, et certains groupes (en particulier en France, ainsi que le groupe de travail de l'IFIP) cherchent d'autres voies, plus méthodologiques. Le principal grief qui est fait est que cette proposition de norme vise à figer les techniques graphiques en leur état actuel, délaissant d'ores et déjà les nouveautés telles que les écrans à balayage télévision. Il est donc probable que la proposition actuelle ne conduira qu'à une norme transitoire, avant l'apparition de logiciels graphiques de haut niveau, proposant en particulier des outils de dialogue d'un niveau convenable.

3. LES APPLICATIONS

Les logiciels d'application se regroupent en quatre grandes catégories :

- les logiciels produisant des dessins ou images statiques
- . logiciels à vocation essentiellement graphique

Il s'agit d'applications dans lesquelles le dessin est le produit final attendu des calculs. Les principaux domaines d'application sont la conception de formes artistiques, la création de motifs pour les industries textiles ou pour la décoration, la composition de textes ou d'affiches, la cartographie.

. logiciels pour la conception assistée par ordinateur
Nous classons dans cette catégorie les applications pour lesquelles le dessin est un support de réflexion et de communication. Le dessin est une représentation des données qui sont traitées, toute modification de ces données étant répercutée sur la représentation et graphique et vice-versa. Les principaux domaines de la conception assistée sont la mécanique, le génie civil, l'électronique, les industries automobile, aéronautique et navale, l'architecture, l'urbanisme.

- les logiciels produisant des dessins ou images dynamiques
La succession des images dans le temps est un paramètre supplémentaire pour ce type de logiciels. Deux catégories peuvent être distinguées :

. logiciels où le temps est simulé
L'enchaînement de la succession des dessins est géré par programme. Nous rangeons dans cette catégorie les systèmes de production de dessins animés et les logiciels d'illustration de programmes.

. logiciels en temps réel
L'enchaînement des dessins se fait en temps réel, commandé par des dispositifs extérieurs au calculateur. Les exemples les plus classiques se trouvent dans le contrôle de processus industriels, le contrôle des trafics urbain, aérien et maritime.

Nous passerons rapidement en revue quelques exemples d'application de chacune des catégories.

Logiciels à vocation essentiellement graphique

Nous commencerons par signaler que quelques artistes utilisent les consoles de visualisation pour leur création, que ce soit pour réaliser des tableaux ou pour préparer des sculptures. Cependant, ces artistes sont à l'heure actuelle fortement handicapés par le bas niveau d'expression des logiciels qui existent. Certaines universités américaines délivrent d'ores et déjà des cours d'art par ordinateur.

Les applications industrielles concernant un aspect un peu artistique se trouvent dans les industries textiles et de décoration. Les motifs de base sont créés directement à l'écran, leur répétition éventuelle étudiée également, jusqu'à la production d'un ruban perforé destiné à conduire des machines à imprimer ou à tisser.

Un axe en fort développement concerne les métiers du livre. L'apparition des photocomposeuses a conduit à des recherches sur des méthodes de création interactive de polices de caractères originales. La mise en page des textes et des schémas se trouve facilitée par l'usage des consoles de visualisation liées à des éditeurs de textes perfectionnés.

Enfin, la cartographie est également en plein essor, les techniques interactives étant utilisées pour l'acquisition et la vérification des données, la création des cartes et du symbolisme utilisé.

Logiciels pour la conception assistée par ordinateur

Il n'est pas possible d'évoquer toutes les applications liées à la conception assistée par ordinateur. Les domaines les plus classiques sont ceux de l'électronique, de la conception d'automobiles, d'avions ou de bateaux ainsi que l'architecture et le génie civil. Cependant, le domaine de la mécanique voit des efforts répétés depuis quelques années.

Les outils de conception en mécanique se trouvent à plusieurs niveaux:

- étude du principe de fonctionnement de mécanisme, à l'aide de schémas fonctionnels et d'études cinématiques,
 - habillage des pièces par dessin des formes,
 - vérification du fonctionnement du mécanisme complet, en particulier en ce qui concerne l'encombrement,
 - visualisation tridimensionnelle des pièces, permettant de vérifier leur ajustement,
 - établissement de la cotation, des plans d'ensemble et de détail.
- On voit donc que l'on cherche à fournir des outils servant aussi bien au dessinateur qu'au bureau d'études.

Les méthodes de calcul aux éléments finis ont également trouvé un support important dans les techniques graphiques interactives, qu'il s'agisse de définir les domaines à étudier et les contraintes appliquées ou qu'il s'agisse d'intervenir dans le déroulement des calculs pour favoriser ou rectifier une solution. Beaucoup de travaux sont réalisés à l'heure actuelle pour la réalisation de logiciels permettant d'utiliser des techniques aux éléments finis sur des objets tridimensionnels. Les principaux problèmes graphiques rencontrés concernent la définition de l'objet tridimensionnel (en général composé de formes élémentaires) et le contrôle de la définition de celui-ci par une visualisation sélective.

Logiciels d'aide à la production de films

Ils correspondent à la catégorie des logiciels permettant de produire des images dynamiques avec un déroulement du temps simulé. On distingue deux types de logiciels :

- les logiciels de production de dessins animés, permettant à un animateur de créer les séquences de dessins directement à l'écran, parfois de les contrôler sur l'écran lui-même, puis obtenir une sortie sur microfilm. Les principaux travaux ont porté sur les techniques permettant de définir un film de manière entièrement graphique, sur les méthodes permettant d'incorporer certaines contraintes aux objets à animer. Nous signalerons que des dessins animés de grande qualité ont pu être obtenus depuis l'apparition des consoles à balayage télévision, conduisant à des films éducatifs de grande qualité. Signalons encore que certains artistes créent leurs films directement sur magnétoscope, disposant alors d'un moyen très souple et très efficace

de création rapide de films aussi bien éducatifs que récréatifs.

- les logiciels pour l'illustration de programmes. Dans ce cas, la différence provient du fait que l'on dispose d'un programme que l'on désire illustrer au cours de son fonctionnement, en vue de contrôler son exécution ou en vue d'expliquer l'algorithme utilisé. Des logiciels ont été mis au point qui prennent entièrement en charge la composition des dessins sur l'écran, à partir de la donnée d'une bibliothèque de représentations standards et de descripteurs permettant d'associer à chaque variable la représentation graphique voulue. Ces logiciels sont opérationnels, mais ne sont pas véritablement exploitables car ils exigent (pour l'instant) une place mémoire considérable. Ils ouvrent cependant une voie dans le domaine des logiciels graphiques intelligents.

Logiciels en temps réel

Ces logiciels sont en général extrêmement spécialisés, destinés à satisfaire une application bien particulière. Les champs d'application les plus classiques concernent le contrôle de processus, le contrôle de trafic urbain, maritime, ferroviaire ou aérien. La caractéristique commune à toutes ces applications est l'utilisation de diagrammes à base de traits, employant quatre ou cinq couleurs pour mettre en valeur les différents phénomènes à surveiller.

Cependant, l'apparition des consoles à balayage télévision a orienté les recherches vers la production en temps réel de paysages, afin de mettre au point des simulateurs de vol pour l'entraînement des pilotes. Les recherches faites à l'heure actuelle mobilisent beaucoup d'énergie. Les objectifs assignés sont d'autant plus ambitieux que les délais donnés aux chercheurs sont en général relativement courts. Ce sera dans ce domaine que les progrès les plus spectaculaires auront lieu dans les prochaines années.

CONCLUSION

Ce rapide tour d'horizon de l'état actuel des recherches dans le domaine des techniques graphiques interactives nous amène à faire plusieurs remarques :

- les progrès réalisés ces dernières années sont spectaculaires, surtout en ce qui concerne la génération d'images réalistes. On peut s'attendre à ce que les algorithmes de synthèse de ces images deviennent assez rapidement relativement communs, permettant d'obtenir des représentations proches de la réalité en des temps de calcul relativement bas.

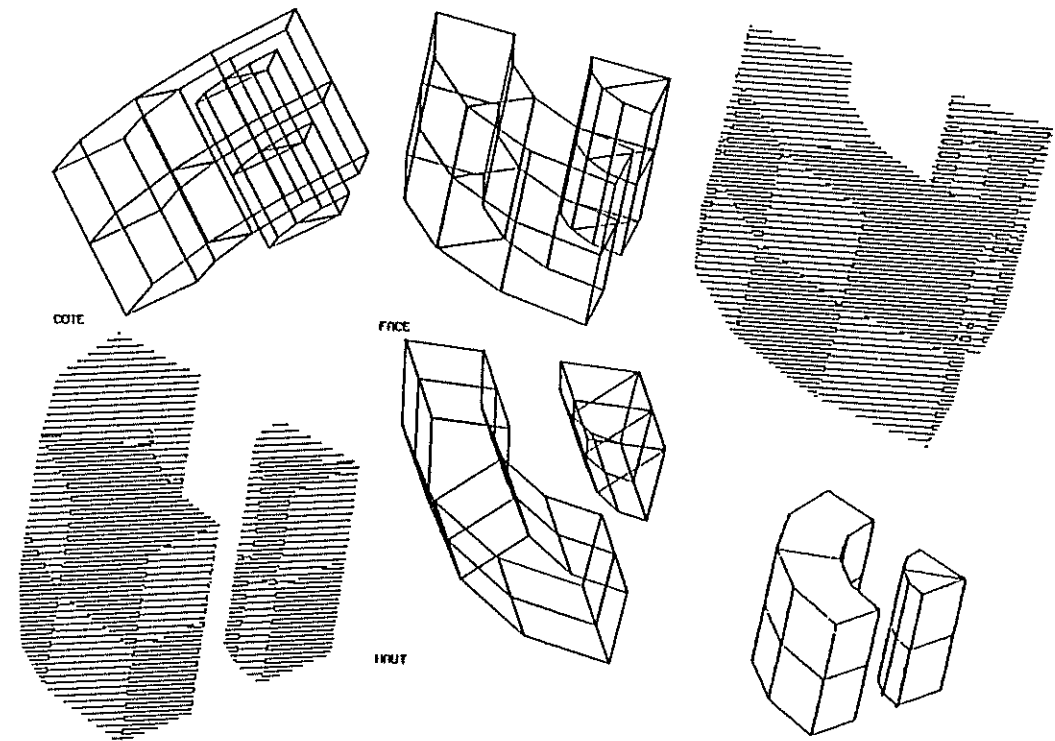
- beaucoup de travail reste à faire en ce qui concerne les techniques d'acquisition de données, point faible du graphique interactif. Le rapprochement constaté avec les chercheurs en traitement d'images et reconnaissance de formes conduira peut-être à de nouvelles voies de recherche qui sont indispensables au développement des techniques de conception assistée par ordinateur.

Nous pensons cependant que les travaux actuels conduiront peu à peu à rendre simple et naturelle la communication graphique avec un ordinateur.

BIBLIOGRAPHIE

1. R.E. BARNHILL, R.F. RIESENFELF, Computer Aided Geometric Design, Academic Press, Londres, 1974.
2. J.F. BLINN, Models of Light-Reflexion for Computer Synthesized Pictures, Computer Graphics, SIGGRAPH ACM vol 11, n° 2, 1977.
3. J.F. BLINN, M.E. NEWELL, Texture and Reflexion in Computer Generated Images, Communications of the ACM, vol 19, n° 10, octobre 1976.
4. I.C. BRAID, Designing with volumes, Cantab Press, Cambridge, 1974.
5. E.E. CATMULL, A Subdivision Algorithm for Computer Display of Curved Surfaces, Ph.D. Thesis, University of Utah, décembre 1974.
6. F.D. CARLSON, Graphics Terminal Requirements for the 1970's, Computer, vol 9, n° 8, août 1976.
7. F.C. CROW, Shadow Algorithms for Computer Graphics, Computer Graphics SIGGRAPH ACM vol 11, n° 2, Summer 1977.
8. S. DAVIS, Computer Data Displays, Prentice Hall, 1969.
9. J. ENCARNACAO, Computer-Graphics : Programmierung und Anwendung von graphischen Systemen, R. OLDENBURG Verlag, Munich, 1975.
10. W. GILLOI, Interactive Computer Graphics. Data structures, algorithm, Languages. Prentice Hall, 1978.
11. GSPC Core definition subgroup, General Methodology and Proposed Standard, Computer Graphics SIGGRAPH ACM, vol 11, n° 3, Fall 1977.
12. R.A. GUEDJ, Report on the IFIP WG 5.2 workshop "Methodology in Computer Graphics", à paraître.
13. M. LUCAS, Contribution à l'étude des techniques de communication graphique avec un ordinateur. Eléments de base des logiciels graphiques interactifs, Thèse d'Etat, Grenoble, décembre 1977.
14. P.MORVAN, M. LUCAS, Images et Ordinateur - Introduction à l'infographie interactive, Larousse, 1976.
15. W. NEWMAN, R.F. SPROULL, Principles of Interactive Computer Graphics, Mc. Graw Hill, New-York, 1973.
16. B.T. PHONG, Illumination for Computer Generated Images, University of Utah, Ph.D. Thesis, août 1973.
17. D.F. ROGERS, J.A. ADAMS, Mathematical Elements for Computer Graphics, Mc. Graw Hill, New-York, 1976.
18. I.E. SUTHERLAND, R.F. SPROULL, R.A. SCHUMACKER, A Characterization of Ten Hidden-Surface Algorithms, ACM Computing Surveys, vol 6, n° 1, Mars 1974.

Ouvrages généraux	9	10	13	14	15	17
Matériel	6	8				
Elimination de parties cachées	5	18				
Ombrage, reflets etc ...	2	3	7	16		
Définition de scènes	1	4				
Normalisation	11	12				



EXPERIENCE WITH A DISTRIBUTED COMPUTING
SYSTEM FOR MAGNETIC FIELD ANALYSIS

M.J. NEWMAN

Rutherford Laboratory, Chilton, Didcot, Oxon, OX11 0QX

ABSTRACT

This paper describes the author's experience over the last two years in developing a computer system called THESEUS. The system is general purpose but the initial use has been for magnetic field analysis. The system involves several computers connected by data links. Some are small computers with interactive graphics facilities and limited analysis capabilities, and others are large computers for batch execution of analysis programs with heavy processor demands. Both the advantages and problems experienced are highlighted, together with a mention of possible future developments.

1. INTRODUCTION

This paper describes a system which has been developed at the Rutherford Laboratory in order to make the most effective use of the different sorts of computer available for the analysis of electro-magnets.

A natural question to ask is why do we need a distributed system, and indeed, in some environments the answer may well be that we do not.

It is desirable for design work to be interactive. Unfortunately, even with the best algorithms available and very fast computers, it may take several hours to analyse a complicated magnet if the non-linear properties of the iron are included. Such problems are obviously not suited to interactive work.

However, it is still an advantage to prepare and verify the data interactively with graphical confirmation, and to display the results graphically after analysis. Most analysis programs check their data for internal logical inconsistencies, but some errors cannot be detected, and it is in these cases that graphical verification is so valuable.

So, some tasks do not make heavy processor demands, but do demand an operating system which supports interactive use with graphics, and vice-versa. If a single computer is available which satisfies all these requirements, then a distributed system will not be necessary. However, we have found it advantageous to select at least two computers, one of which is intended for the interactive work and the other for the analysis. This distinction has already been weakened by the use of the interactive computers for some light analysis tasks, and we expect that future developments will weaken this distinction even further.

2. THE THESEUS SYSTEM

The THESEUS system is a suite of three-dimensional interactive graphics programs which assist engineers in the preparation of data for finite element programs and in the interpretation of results. An early version has been described in a previous paper¹. The analysis program to which it has been lined for magnet design is GFUN^{2,3,4,5}, but it has been designed so that it can be easily used with any similar analysis program.

The GFUN program and the parts of the THESEUS system which are particular to GFUN run in either of a pair of IBM 360/195 computers. The interactive part of THESEUS runs in a PRIME 400 computer or a GEC 4070 computer. The two computers containing the two parts (batch and interactive) of the THESEUS/GFUN system are connected by a direct link, which is used for job submission and output retrieval. Figure 1 shows diagrammatically how the system works.

When the data on the PRIME computer is ready for analysis, the appropriate files are concatenated and sandwiched by JCL for the IBM machine, and the resulting file is submitted to the IBM computer for a normal batch run. The first program to run in the IBM is a pre-processor for GFUN. This program takes as input the file of data in the THESEUS format and translates it into a temporary file of card images in exactly the format required by GFUN. This has the effect of a data explosion by typically a factor of 15.

The GFUN program will then run, read these data card images and the normal lineprinter output will also be placed on a temporary file for subsequent interpretation by a GFUN post-processor. The post-processor for GFUN has not yet been written. This will translate the output from GFUN into the format of results data expected by THESEUS. This file of results will finally be sent back to the PRIME computer for interactive graphical inspection using THESEUS.

One advantage of this system is that the analysis program need not be modified in any way. In fact, in order to interface THESEUS to another similar analysis program, only the immediate pre- and post-processor would need to be re-written.

The suite of interactive programs interface to each other by means of files organised in a common database. Programs are available to perform several well defined general tasks. For example, overall geometry definition, automatic mesh generation, material properties specification, etc. The system is highly modular and can be easily extended by adding new mesh generators etc.

3. GEOMETRY DEFINITION PROGRAM

In order to analyse a given magnet design the user may wish to make several finite element models of varying complexity, but for each of these the overall geometry of the design will remain fixed. For this reason, central to the structure of the THESEUS system is a file which describes the overall geometry of each design without

any reference to how that geometry is to be sub-divided into finite elements. In principle, any number of geometry definition programs can be provided in the system, each of which would produce a geometry file in the standard specified format. Also, any number of different mesh generator programs may be provided which take the standard geometry file as input and sub-divide it under user control. In fact, at present the system contains only one geometry definition program and one mesh generator program. This geometry definition program provides the user with a simple keyboard based language in which he can define a wide range of three-dimensional solids. For magnet design, the exact dimensions of iron and conductors are often critical, and it is necessary to type the numbers in on a keyboard. For other applications, a digitiser may be sufficiently accurate and more convenient. A suitable geometry definition program, using a digitiser for input and providing a standard geometry file as output, could be added to the suite of programs.

As each item of geometry data is defined the program responds with graphical verification on the screen, and the user may cancel the item or proceed with the next.

For several reasons, the user is asked to divide his general three-dimensional solid conceptually into smaller regions each of which is taken from a set of primitive shapes (Fig 2) which the program recognises. This makes it easier for the user to envisage the shapes he is defining, and because they will often contain a degree of symmetry, the amount of data needed to define them is reduced. Furthermore, such a sub-division will be necessary anyway whenever the total solid is comprised of regions with different properties (eg. different permeability curves). The sub-division may also mirror the way in which the object will eventually be constructed from simpler parts.

Each region is defined separately, the appropriate primitive shape being identified by a name such as BRICK or WEDGE. Each region can be given several attributes with default values assumed if not supplied. Each region can be either a conductor or a non-conductor (iron). Each can be given a four character symbolic name. Regions with identical properties (permeability) should be given the same symbolic name. This name will be mapped onto a file containing a table of these properties when the magnet is analysed. In this way the effect of several different permeability steels can easily be assessed. Various reflection codes can be associated with each region so that for magnets with symmetry, the amount of data can be reduced. For conducting regions, a symbolic name can be supplied for a current density which will be replaced by an actual value when the magnet is analysed.

The detailed geometry of each region is defined using a special language called POLLY. This is based on six single letter codes L, R, U, D, I, O meaning 'left', 'right', 'up', 'down', 'in' and 'out'. By means of these associated with numerical values and combined as necessary it is possible to define any vector. Each

code letter is followed by a number which is an incremental change in the appropriate coordinate. The program can be thought of as providing an instrument for drawing in three-dimensional space. The first mode must be specified by three coordinate values, but subsequent movements may be specified in several ways; incremental movements using the letter codes and a number; movements to absolute coordinates using the letter code plus a 'T' and a number; and by specifying all three coordinates. Since many edges will be parallel to one of the principal axes, only one code need be supplied. For example, to define an orthogonal brick the following data could be provided. The program knows that the opposite faces of a brick are parallel.

Data	Meaning	Program Response
10 10 0	coordinates of first node	puts a cross at this point
R80	first edge is 80 units in right direction	draws vector 80 units to right
U10	next edge is 10 units up	draws a vector 10 units up and adds 2 more edges parallel to first two to complete a rectangle. (current pen position at 1st node)
O100	next edge is 100 units out	draws this edge and completes drawing of brick.

Thus a minimum of data is required and an immediate visual check can be made for errors. At any stage, if the user detects an error in his data he can cancel the last command. Only limited curve facilities are provided at present in the form of 90 degree arcs lying in one of the principal planes. The set of primitive shapes can, however, easily be extended. Perspection views with hidden lines removed or drawn dotted can be provided. (Fig. 3 and 4).

4. AUTOMATIC MESH GENERATOR PROGRAM

The THESEUS system is designed to allow any number of mesh generator programs to be incorporated, each being easily interfaced to the standard geometry data file in the common database. The only mesh generator which has been implemented so far is the one which has proved most useful to the GFUN program. This allows a general hexahedron to be sub-divided into smaller hexahedra under user control, and each of these to be filled automatically with tetrahedra. As each region is displayed, the origin and three basis directions are labelled and the user can specify how many sub-divisions are made in each direction. Figure 5 shows a typical resulting set of tetrahedra.

The mesh is generated by setting up a polynomial mapping onto a unit cube, sub-dividing this in a regular fashion, and inverting the mapping to find the actual mesh coordinates. This method is cheap to compute and is effective provided that the hexahedron is not too drastically distorted from the cube.

Although the complete detail of the mesh may be displayed on the graphics screen, only the few commands needed to define this subdivision are stored in the associated mesh file. In this way, the number of elements which can be generated is unlimited. The actual explosion of data occurs in the pre-processor program which runs in the large batch computer. This contains the vital subroutines from the mesh generator, and outputs the appropriate card images for GFUN instead of displaying the mesh on a screen.

5. MATERIAL PROPERTIES DEFINITION PROGRAM

This program allows the user to define tables of B and H values representing the permeability curves of iron. He can display these curves and store them in the database for subsequent inclusion in the data for a magnet design analysis.

6. BATCH JOB SUBMISSION

The profile selection program allows the user to specify which geometry file he wishes to analyse, and, for that geometry, which of possibly several mesh files he wishes to use to generate the finite element model. He also specifies which magnetic materials he wishes to use and what actual current densities he wants in the conductors. The program automatically concatenates the appropriate files, sandwiches them with JCL for the batch machine, and submits the job.

7. GRAPHICAL INTERPRETATION OF RESULTS

The post-processor for GFUN has not yet been written, and so this part of the system is not yet complete. However, standard utilities for displaying tables, graphs, and contour maps do exist see Figs. 6 and 7. A package of subroutines has been written which allows the easy manipulation of tables representing real functions of any number of variables. In addition to handling tables of the three components of field strength over a three-dimensional rectangular grid it is also used for the tables representing permeability curves.

8. SOME SPECIAL FEATURES

8.1 Linked list data structure

This package of routines allows a program to store data in a single large one-dimensional array in a structured manner as linked lists without needing to bother with the details of storage management. The array is divided into cells of equal length which are placed on a stack. This avoids garbage collection problems. This method of data structure means that the programs can accommodate with equal

ease a large number of simple shapes or a smaller number of more complex shapes. Editing is also much more efficient as only pointers need to be changed.

8.2 Command syntax analyser

Command names and parameter names may be abbreviated so long as they remain unambiguous. Syntax errors are detected and explicit error diagnostic messages are immediately sent to the terminal. The programs are driven by the user selecting a command from a menu rather than by the programs prompting the user for a reply. This is essential to avoid frustrating experienced users. Inexperienced users can obtain on-line documentation both from the command syntax analyser and from a file of documentation. The analyser performs all numerical formatting so that all input and output is in AI format.

8.3 Portability and Device independence

Almost all of the code is written in ANSI x 3.9 - 1966 Fortran. Where a knowledge of the machine hardware or operating system is unavoidable, interfaces to machine dependent subroutines have been defined. Examples are character packing and filing operations.

The GINO-F graphics package is used in storage tube mode and each new device needs only one specific sub-routine to be written.

8.4 Application independence

Most of the system is unaware whether the object concerned is a magnet whose field is to be analysed or a bridge whose stress is to be calculated.

If several different analysis programs are to be compared, once the pre- and post-processors are written, the same problem can be run on each very trivially.

9. GENERAL PURPOSE DATABASE

Due to the modularity of THESEUS it is necessary for several files to be stored during the analysis of a design. Without a convenient form of structuring these files according to their logical relationships, the system would rapidly become unusable. The database system which has been written for THESEUS provides this framework and is the common link between all parts of the system. It can also be shared by systems other than THESEUS. It is developed in several logical layers of software, which, at the lowest level, calls the manufacturer's system subroutines to perform basic operations such as creating, closing, and deleting files. At one level it assumes that the filing system on the computer allows a tree structure of directories. If this were not the case, this could be simulated by software operating on a single large direct access file.

The database provides several facilities which are not yet normally

found in manufacturer's filing systems.

Firstly, it does not distinguish between data files and directories. Each file is both. It can have data stored in it, and other files can be attached to it at a lower level in the database. Thus, a geometry file contains the data defining the overall geometry of a design and has attached to it each file which specifies a finite element discretisation of that geometry. Lower down the tree will be found profile files, data files, and results files all structured in the obvious manner.

Secondly, the database provides protection facilities not normally available in the filing system. In addition to each file having an individual owner (ie. the person logged in to the database when the file was created), each file has a program owner. Thus files cannot accidentally be deleted by the wrong program and the integrity of the system is preserved. Four levels of ownership can be obtained for both individuals and programs. These are owner, key-holder, partner, and non-owner.

Thirdly, although this feature has not yet been implemented, the database can set up relationships between any of its files. For example, a useful relationship would be "Do not delete file A as it is referred to as data by file B". Then, when a file of perm-eability data had been used as data in a magnet design the owner would be prevented from deleting it until the design had been deleted.

10. IMPLEMENTATION PROBLEMS

The interactive programs in the THESEUS system are large programs in mini-computer terms. They may have 12000 cards of source and occupy 200 Kbytes of memory. They have been designed to be portable, modular, and easy to maintain in the expectation that these factors will be more important than small memory requirements. This does mean that a virtual memory system is necessary in a small computer. The PRIME 400 computer has 0.75 Mbytes of memory and 760 Mbytes of disc space. The paging architecture makes it very easy to develop large programs. Early experience with the GEC 4070 computer, which allows segments of varying size, indicated that it was necessary for the user to spend some time specifying the segmentation structure in order to achieve a performance which approached the optimum. The system software has since been improved and we do not have any recent experience on this machine.

11. INTERACTIVE ANALYSIS

We have experimented with the PRIME 400 and GEC 4070 computers for limited field calculations. At present we can cater for two-dimensional problems without iron. The algorithms used are those from GFUN. As an example, Figs.6 and 7 show the fields produced by a dipole with curved sides. The calculation of 400 field points took 100 seconds on the PRIME.

12. FUTURE DEVELOPMENTS

The most interesting development for magnet analysis will be the addition of an array processor (F P Systems AP 120 B) to the PRIME computer. This will provide approximately a factor of 10 improvement in speed for problems which can be programmed to take advantage of the special hardware. The hardware should arrive within two months, and we are considering different programming techniques.

Another interesting development is the reduction in cost of raster scan graphics devices with colour. We have a colour device with 256 x 256 resolution for the same cost as a middle range storage tube device. We are also experimenting with this to see how it can be exploited for magnet design.

13. CONCLUSIONS

The paper describes a flexible software system spreading across a range of computer hardware. The system is highly modular for easy extension and highly portable for transfer to different computers. It can easily be adapted for a completely different application. It provides a highly efficient and flexible interface between magnet designers and specialised analysis programs.

14. REFERENCES

1. Newman, M J. 'Application of Interactive Graphics Techniques to Magnet Design'. Proc COMPUMAG Conf. on the Computation of Magnetic Fields, Oxford 1976.
2. Newman, M J, Trowbridge, C W, and Turner, L R. "GFUN: An Interactive Program as an aid to Magnet Design". Proc 4th Int.Conf. on Magnet Technology, Brookhaven, 1972
3. Newman, M J, Simkin, J, Trowbridge, C W, Turner, L R. "GFUN User's Guide - a User Guide to an Interactive Graphics Program for the Computer-Aided Design of Magnets" RHEL/R244, 1972.
4. Collie, C J, Diserens, N J, Newman, M J, Trowbridge, C W. "Progress in the Development of an Interactive Computer Program for Magnetic Field Design and Analysis in Two and Three Dimensions". RL-73-077, 1973.
5. Armstrong, A G A M, Collie, C J, Diserens, N J, Newman, M J, Simkin, J, Trowbridge, C W. "New developments in the Magnet Design Computer Program GFUN". RL-75-066, 1975.

15. ACKNOWLEDGEMENTS

The author would like to acknowledge the support of C W Trowbridge and the help of several systems programmers throughout this work.

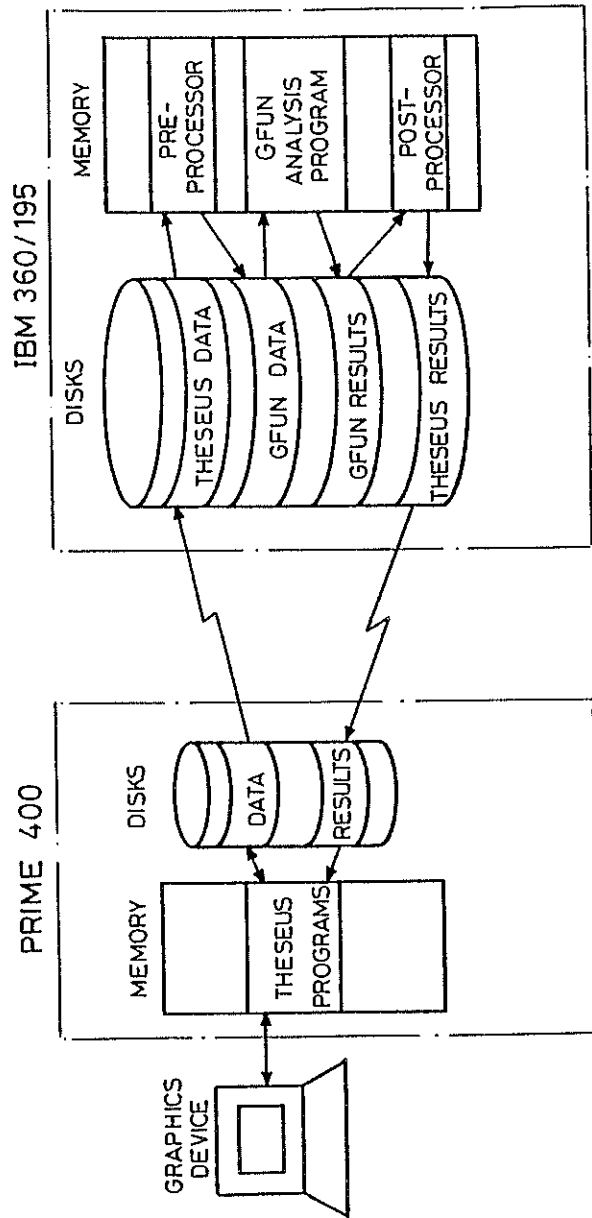


FIG. 1 - BLOCK DIAGRAM OF SYSTEM

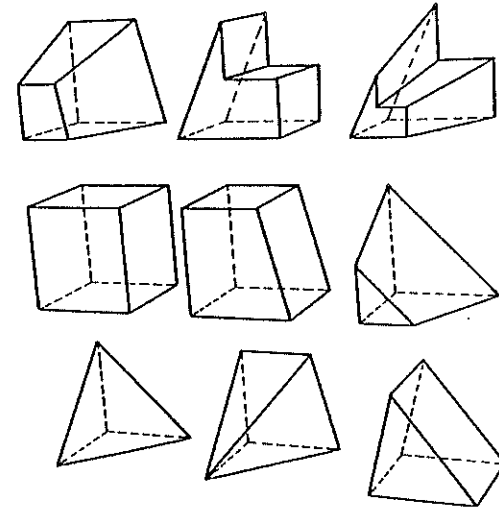


FIG. 2 - 3D PRIMITIVE SHAPES

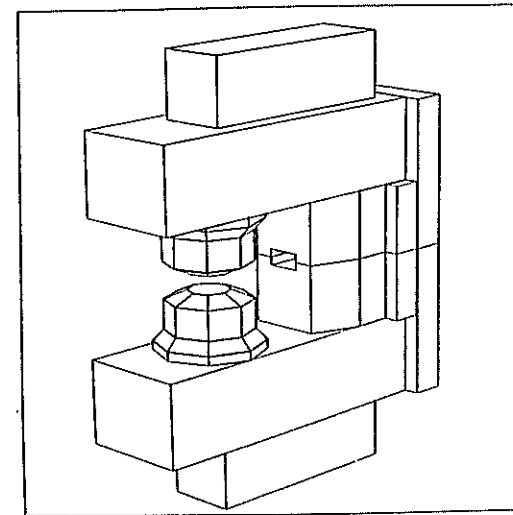


FIG 3 - PERSPECTIVE VIEW WITH HIDDEN LINES REMOVED

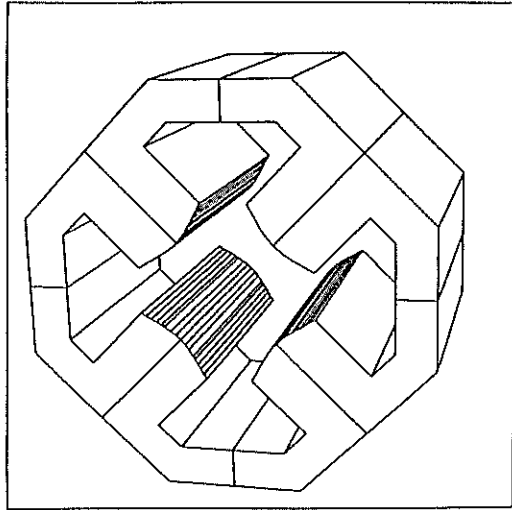


FIG. 4 - QUADRUPOLE MAGNET

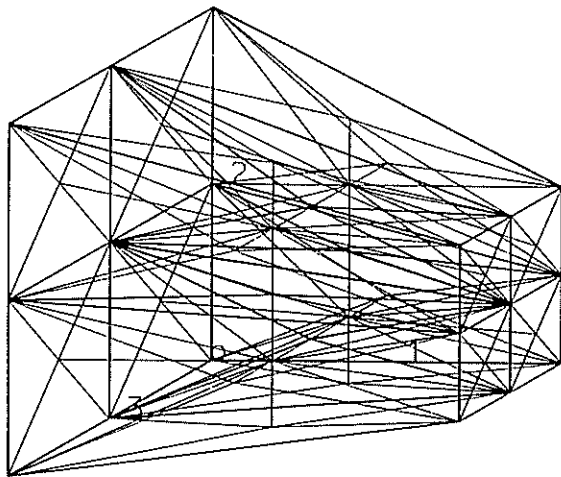


FIG. 5 - TETRAHEDAL MESH

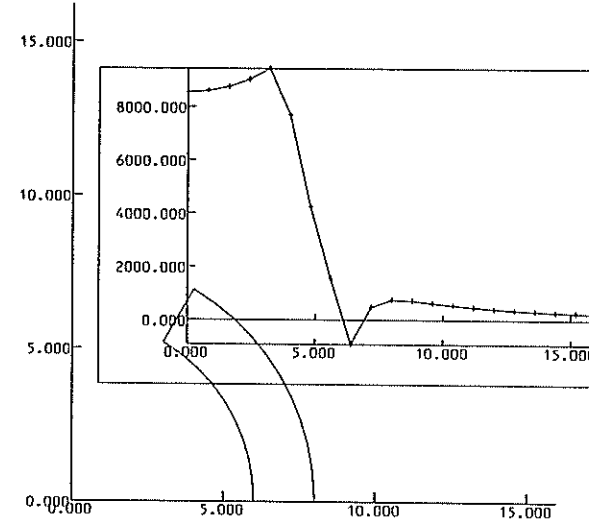


FIG. 6 - GRAPH PLOT OF FIELD

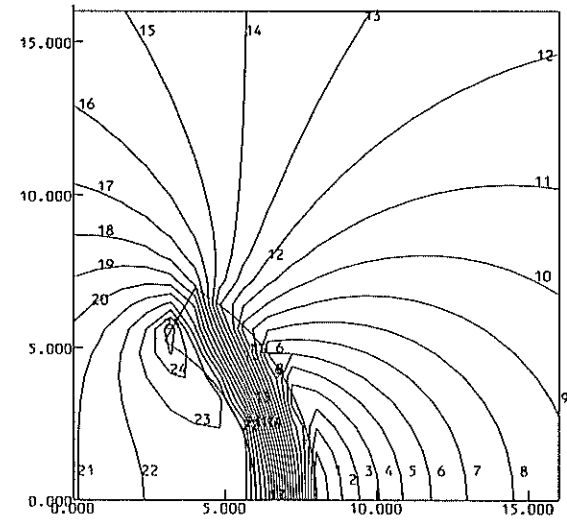


FIG. 7 - CONTOUR MAP OF FIELD

DEFINITION AND DRAWING OF ELECTROTECHNICAL DEVICES
FOR FIELD CALCULATION.

ANCELLE B.
Laboratoire d'Electrotechnique
E.N.S.E.G.P. Grenoble FRANCE

ABSTRACT

We have developed a computer interactive program with which we can design an electrical device using either simple geometrical elements or complex one. These complex elements have a tree structure which gives an easy way to the user to build a device. Some geometrical transformations such as translation, rotation, symmetry and general scaling allow the user to easily manipulate every element. At each stage of the design the user can display either a part or the entire device on a graphic terminal screen. The drawn picture is a simple perspective projection on an image plane chosen by the user. After the geometrical design is completed, the user can define a subdivision of the boundaries of the object. Therefore this program can give input data for a field calculation program with BIE method.

1. INTRODUCTION

In this paper we will focus our attention on the definition of input data for field analysis program. Prof. Silvester(1) with his MAGNET system, M. Newman(2) with GFUN program have shown the capabilities of interactive graphics. We have developed an interactive program with which we can define any electrical device. It has 2 steps : first the definition and drawing of the device, second the definition of the physical properties and the creation of a mesh model. We will emphasize on the first step. The program is an interpreter which uses the ENTREE language and has been designed for building a complex structure from a minimum number of explicitly defined objects. The electrical devices are only defined by their geometrical form and the program allows for tree-structured objects. The data base is composed of the coordinates of the points and of a set of topological elements. These two items are referenced by a number. The system owns also a library of models in which standard models can be stored, and the user can easily retrieve them when he needs them. The models are referenced by their name in the library.

2. THE DEFINITION AND DRAWING MONITOR

The program is composed of two parts : first the monitor modules which handle the ENTREE language and second the execution modules which process the commands. We will first describe the monitor and its commands and afterwards study the execution modules and their capabilities.

2.1 MONITOR COMMANDS

The monitor modules read the commands from the keyboard of the terminal, interpret them and call for the right execution module. The monitor modules are composed of a multidirectional switch and a man machine exchange set. Every command is read from the keyboard of the terminal and has the following format:

COMM NNN TT DT

where COMM is the name of the command, NNN is a number (point or element reference), TT and DT are options. The commands names have four characters but the main commands are abbreviated by only one character (underlined character on fig. 1). When NNN is either a point or an element reference and its value is zero the command acts on the whole object. Fig. 1 gives the directory of commands.

The LIT command requires the user to give the data defining the object. The user has to type the coordinates of the points and the components of topological elements at the keyboard. Direct graphic input using either the cursor or an analog tablet is under development. The MODI command may be used to correct an error in the data. The option identifies the type of false data. If the user wants to delete either a point or an element, he has to type the command DELP or DELE respectively. At the beginning of a session the user has to give the dimension (2D or 3D) of the object thanks to the DIME command. The next TRAN, ROTA, SYN, SYMX, SYHY, SYHZ, GROS, POS commands either generate a new element or redefine an existing one using geometrical transformations. The MAC command may be used to define an assembly of elements. The reference numbers of points or elements can be displayed using the IDEN command. The ZERO command clears the data base and initializes the program. The AFFI command displays the object or the NNN element on the screen. For three-dimensional objects the user can give as parameters the kind of projection he wants : either an orthographic projection

Fig 1 Monitor Commands

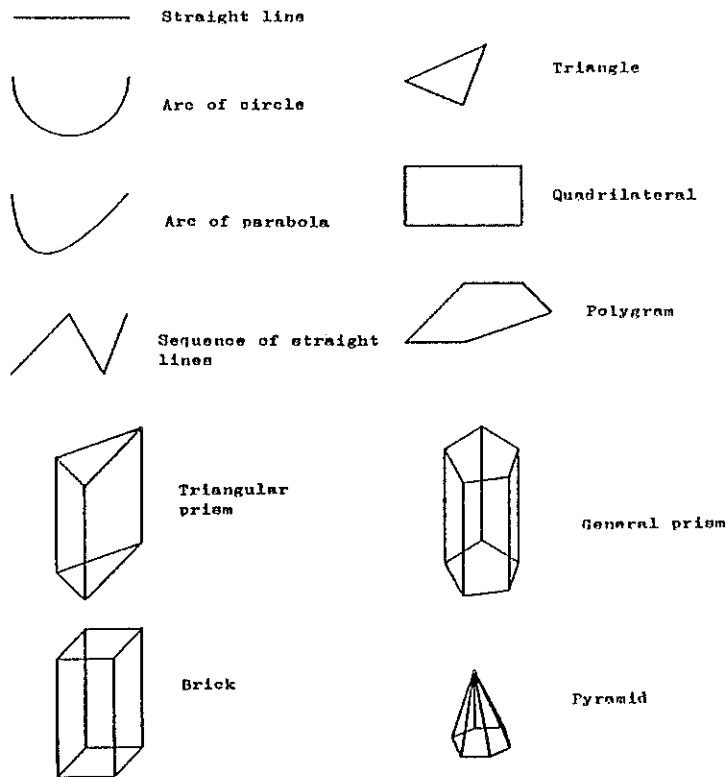
commands	options	
LIT		Definition of the object(coordinates and geometrical elements).
MODI	PT EL	Correction of an error in the data.
DELE	DT	Deletion of an element ; if the option DT is given the whole components of the element are deleted.
DELP		Deletion of a point.
DIME		Definition of the dimension of the object.
TRAN	ME	Translation of an element.
ROTA	PT ME AX	Rotation of an element around a point if the option PT is given or rotation around an axis if the option AX is given.
SYM	ME	Symmetry about a plane.
SYMX	ME	" " " " : x = a
SYMY	ME	" " " " : y = b
SYMZ	ME	" " " " : z = c
GROS	ME	General coordinates scaling.
POS	ME	Composition of geometrical transformations.
MAC		Creation of an assembly of elements.
IDEN	EL PT	Display the references on the screen.
ZERO		Clearing and initialization.
AFFI		Display of an element.
DESS		Drawing on a plotter.
ZOOM		Zooming of a part of a picture.
EFFA		Erases the screen.
ECRI		Prints the content of the core.
CONT		Displays the content of the core.
FENE		Modifies the size of the screen window.
INIT		Clears and initializes the library.
DIR		Gives the directory of the library.
SAVE		Stores data into the library.
CHAR		Loads data from the library.
DEFI		Deletes a member of the library.
SQUE		Squeezes the library.
DEC		Generates the mesh model.
BOB		Defines the windings of the device.
IMP		Stores input data for field analysis.
FIN		Terminates the session.

or an isometric one. The ECRI and CONT commands may be used to check the content of the core. Generally the mapping from the user's reference coordinate system onto the screen of the terminal maximizes the image size on the screen, however the user can choose the size of the window using the FENE command. The next six commands are used to manage the library of models. The first one INIT clears and initializes the library. The DIR command gives the directory of the library (the names of all the models, the room they occupy and the room which is free). The SAVE command stores the active data base into the library. The CHAR command loads a model from the library into the core. The DEFI command deletes a member of the library and the SQUE command squeezes the library. The DEC command generates the subdivision of the boundaries of the object for the Boundary Integral Equations Method. The BOB command defines the windings of the device and reads the values of the currents, and the IMP command stores the input data for field analysis into a file. Because of the modular structure of the program new commands can be easily defined and added to the directory. We can notice that the ENTREE language has no macro-capabilities except the POS command, but an editor extension is planned.

2.2 DATA BASE

The data base contains information which give the structure of the device. This structure is well defined but the description of the device is not unique. For many devices a good way to define them is to split them into parts and to split the parts into elementary pieces (the decomposition can have more than 2 levels): only a few number of elementary objects has to be explicitly defined and the connections between parts are easy to describe (3,4). This partition of the device involves the capability of the data base to contain images of tree-structures. There are 2 kinds of elements in the data base: first the simple elements which represent elementary geometrical shapes and second the macro-elements which define the union of other elements. The simple elements listed in fig. 2 are composed of linear elements, surface elements and volume elements. The glossary of simple elements is easily extensible. As example an electric relay is given in fig. 3. It is built in three stages. The core (3), the magnetic circuit (4), an half coil (1) and the movable part (6) are the only explicitly defined objects of the device. In the second and third stages, the coil, the fixed part and the relay are built as assemblies of explicit parts. Another example is given in fig. 4.

Fig. 2 Directory of geometrical elements.



2.3 GEOMETRICAL TRANSFORMATIONS

The user may apply any of the 4 next geometrical transformations: translation, rotation (either about a point or about an axis), symmetry and general scaling, to any element of the data base. He can build new elements and add them to the data base and he can also redefine the elements he transforms. In the first case the program creates new points and new elements with the same structure as the previous elements have: if the user transforms a macro-element then the new elements will be composed of a macro-element and all its components. After the transformation is completed the program looks after the double points and keeps only single points. In the second case only the coordinates of the points change. The parameters of a transformation are asked in the line following the command:

- translation
XTRAN YTRAN ZTRAN
- rotation about an axis
ANGLE XPOINT YPOINT ZPOINT LATITUDE LONGITUDE
- rotation about a point
ROTATION PRECESSION OWN ROTATION
XPOINT YPOINT ZPOINT
- symmetry about a plane
XPOINT YPOINT ZPOINT LATITUDE LONGITUDE
- general scaling
XPOINT YPOINT ZPOINT XSCALE YSCALE ZSCALE

We can notice that the direction of an axis is only given by its longitude and latitude angles. All calculations are processed with (4x4) matrices because we implicitly use homogeneous coordinates(2). It allows the program to have a very simple structure and to perform easily the composition of transformations. In this later case the names of the transformations (up to 3 different transformations) are typed on the first line of parameters and the next lines contain the numerical arguments of each transformation. Fig. 5 and 6 gives some examples of transformation.

2.4 DRAWING OF THE DEVICE

At any time of the session the user can display a part or the entire object he has built. For three-dimensionnal objects the

Fig. 3 Relay

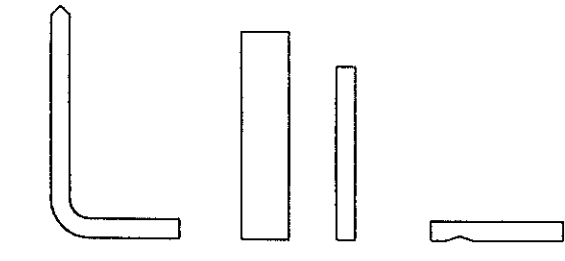
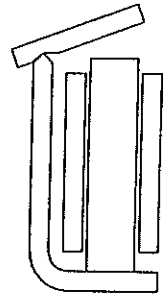
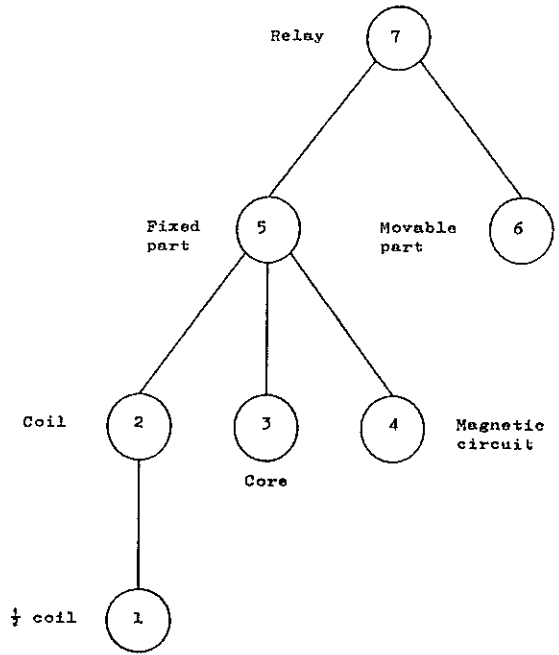


Fig.4 Rotor

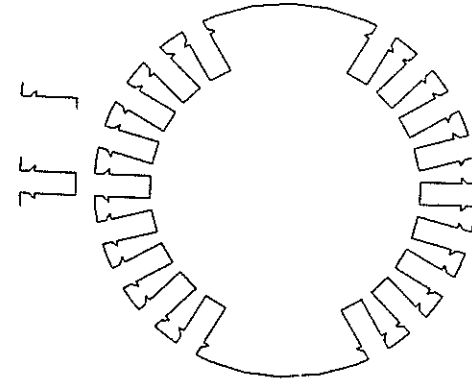
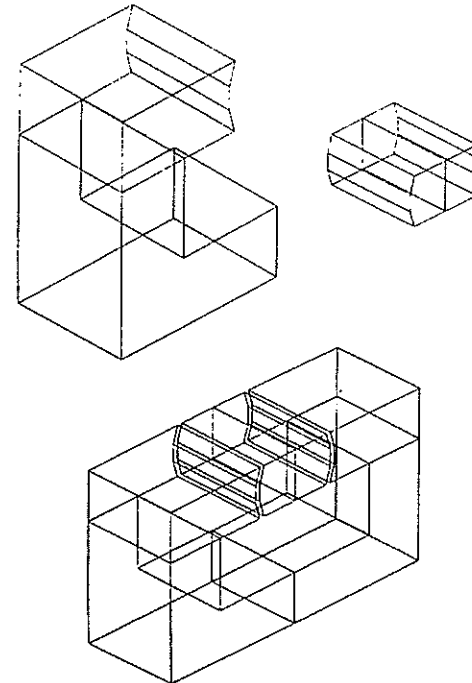


Fig.5 Relay



user can choose the projection he wants either orthographic or isometric projection but now hidden lines are not removed. In the first case the six views of mechanical drawing can be displayed : the front view, the right view, the left view, the top view, the bottom view and the rear view. In the second case eight isometric projections can be displayed, each of them showing a distinct corner of the object : the upper right front corner, the upper left front corner, the upper right rear corner, the upper left rear corner, the lower right front corner, the lower left front corner, the lower right rear corner, the lower left rear corner (fig. 7 and 8). In each case the program automatically maximizes the size of the picture on the screen but the user can easily change the size of the window.

3. PREPARATION OF INPUT DATA FOR FIELD CALCULATION

After the definition of the object is completed the user can define a subdivision of the boundaries of the object for FIE method and enter material properties and the boundary conditions. Interactive modules ask for these data and process the subdivision of the simple geometrical elements. At the present time only 2D subdivision modules are implemented. The preparation of input data for field calculation is split into two stages. In a first time a regular subdivision of the linear geometrical elements is created and in a second time the topological description of each material region is automatically defined. In this description the subdivision points are automatically ordered counterclockwise to secure data against error. Afterwards data are stored in a file in order to be ready to be processed by the analysis program. The 3D subdivision which is in progress is composed of 3 stages. The program first looks for common sides and locates them. In a second time a regular subdivision of the sides of elements is created and in the third step the program automatically defines the description of the regions.

4. CONCLUSION

We have developed an interactive program to define electrical devices and to draw them. This program is only the first link of an electrical devices design system. We have outlined its main capabilities which are :

- the capability of handling tree structured devices,
- the capability of displaying significant pictures,
- the capability of piling up models which are immediately available.

However the program has some limitation. The main one is the absence of the hidden lines removal. We have in mind the implementation of some available algorithms and the implementation of macro-commands and graphical input.

The author would like to thank Mr. Gauthier for his programming help in this work.

REFERENCES

- 1 P. P. SILVESTER : A postprocessor for the MAGNET system. Internal report. McGill University, Montreal, 1976.
- 2 M. J. NEWMAN : Application of interactive graphics techniques to magnet design, RL 76 027, Rutherford Lab, Oxford, 1976.
- 3 J. R. WARNER : Design applications of the MIDAS graphics system. Computer and Graphics, Vol 2, 1976.
- 4 F. BERNARD : GEDVA : un systeme geometrique tridimensionnel de conception des volumes. Mecanique-Materiaux et Electricite, n 335, Nov 1977.
- 5 D. F. ROGERS, J. A. ADAMS : Mathematical elements for computer graphics. Mc Graw Hill. New york. 1976.
- 6 J. H. BIFFLE, H. A. SUMLIN : Three-dimensional structural analysis using interactive graphics. Computer and Graphics. Vol 2. 1977.
- 7 J. J. BAULTE : Conception assistee par ordinateur de table de pieces mecaniques a caractere fortement bidimensionnel. Mecanique-Materiaux et Electricite. n 336. Dec 1977.

Fig.6 Rotation about an axis

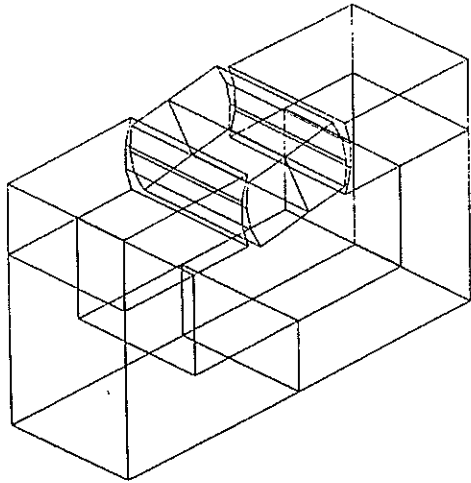


Fig.8 Relay

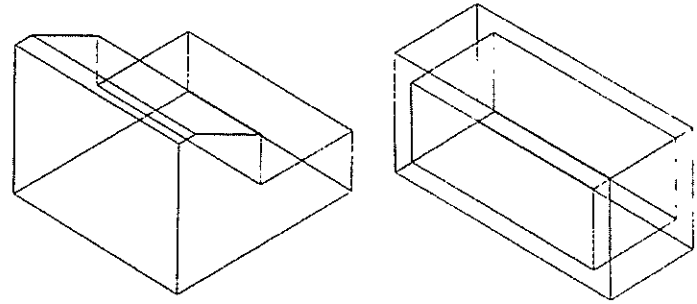
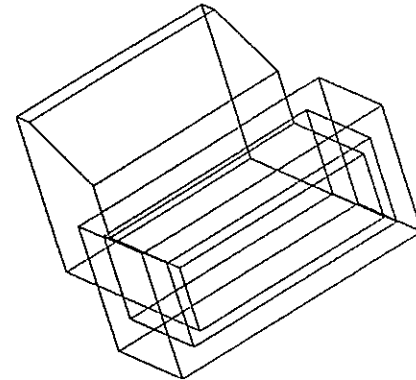
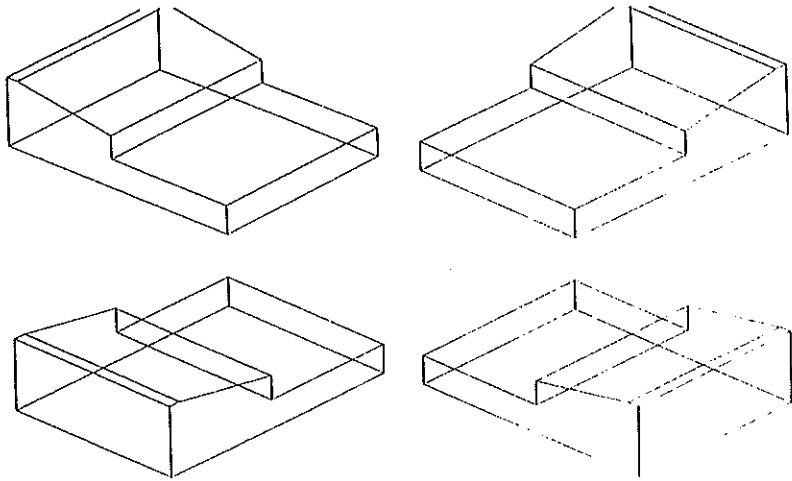


Fig.7 Isometric projections



DEVELOPMENTS OF THE EDDY-CURRENT
PROGRAM EDDYNET*

L. R. Turner and R. J. Lari
Argonne National Laboratory
9700 S. Cass Ave.
Argonne, Illinois 60439 U. S. A.

ABSTRACT

The program EDDYNET solves eddy current problems by means of an integral equation approach. The conducting material is represented by a network of current-carrying line elements. Consequently, Maxwell's field equations can be replaced by Kirchhoff's circuit rules.

The loop equations for voltages, supplemented by the node equations for the currents, comprise a set of linear equations that can be solved repeatedly to give the time development of the eddy currents. Currents, magnetic fields, and power are calculated at each step.

Two further developments to EDDYNET are reported. More complex geometries, with more elements, can now be solved because a TRIM-like mesh generator and internal indexing of lines, nodes, and loops have been incorporated. Also, results can now be in the form of movies representing currents, field penetration, and power distribution. Quantitative comparisons with analytical calculations are presented; movies are described.

1. INTRODUCTION

In designing the magnet systems for fusion reactors and in many other applications there is need to predict the forces, torques, power dissipation, and field distortion resulting from eddy currents. Analytic expressions are available only for the simplest geometries; most situations call for computer calculations. Available programs have been reviewed by Carpenter,¹ and several were described at the 1976 COMPUMAG Conference.

The integral equation approach, which has been applied successfully to magnetostatic problems via the program GFUN,² seems especially applicable to eddy current problems; with it one can treat disjoint bodies without having to include the space between and around them in the calculation. Biddlecomb et al.³ have reported a two-dimensional integral equation method using scalar and vector potentials.

In the program EDDYNET,⁴ the conducting materials are represented by a network of current-carrying line elements. This representation leads to the simplification that Maxwell's field equations can be replaced by Kirchhoff's circuit rules. The loop and node equations can be solved repeatedly to give the time development of the eddy currents.

This paper reports two further developments of the program. First, more complex geometries, with more elements, can now be solved because a TRIM-like mesh generator is used, and the internal indexing of lines,

* Work supported by the U. S. Department of Energy.

nodes, and loops have been incorporated. Second, interpretation of the solutions is facilitated through output in the form of movies showing time-variation of field penetration and current.

The EDDYNET model is reviewed in the next section. The third section describes the computer programs, and the fourth presents results using many elements and describes the movies which have been made.

2. REVIEW OF THE MODEL

In most of the applications described below, the model has been applied to infinitely long prisms and cylinders, but for simplicity in this section we consider the conductor to be a plane figure, of arbitrary shape but simply connected, perpendicular to the applied field. The figure is divided into a triangular mesh; the network of mesh lines is used to represent the conductor. The network contains ℓ lines, m loops, and n nodes; ℓ , m , and n obey

$$m + n = \ell + 1. \quad (1)$$

For each node, Kirchhoff's node rule holds; the algebraic sum of the currents into each node is zero. If the entire conductor is considered, i.e., if symmetry considerations are not explicitly incorporated, then the overall conservation of the current makes one of these n equations redundant; there are $n - 1$ independent node equations.

For each loop, Kirchhoff's loop rule holds; the sum of the IR voltage drops around each loop equals the emf of the loop. The emf is due to the change in applied flux through the loop plus the change in the flux due to all ℓ currents. The flux through loop j due to the current in line i is given approximately by $G_{ji} I_i$ where I_i is the current in line i and,

$$G_{ji} = \frac{\mu_0 A_j}{4\pi r} \left[\frac{gL}{r_2} + \frac{(1-g)L}{r_1} \right] \quad (2)$$

with A_j the area of loop j and r , g , L , r_1 , and r_2 , as defined in Fig. 1. Equation (2) is based upon the approximation that the flux through loop j equals the area times the field at the centroid of loop j .

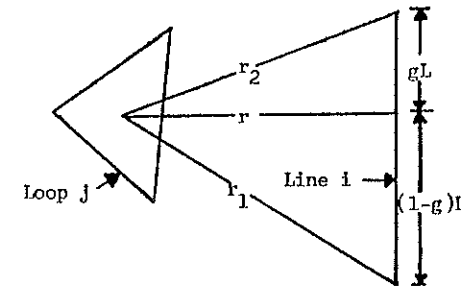


Fig. 1. Parameters to find flux in loop j due to current in line i .

For an infinitely long prism or cylinder, I_i represents the current per unit length, and Eq. (2) is replaced by

$$G_{ji} = \frac{\mu_0}{2\pi} \Lambda_j \left[\tan^{-1} \frac{gL}{r} + \tan^{-1} \frac{(1-g)L}{r} \right]. \quad (3)$$

By Eq. (1) the total number of equations $m + n - 1 = \ell$, the number of unknown currents. Figure 2 shows one-eighth of a square centered on the origin. The one-eighth square has 16 sides, 8 loops, and 9 nodes.

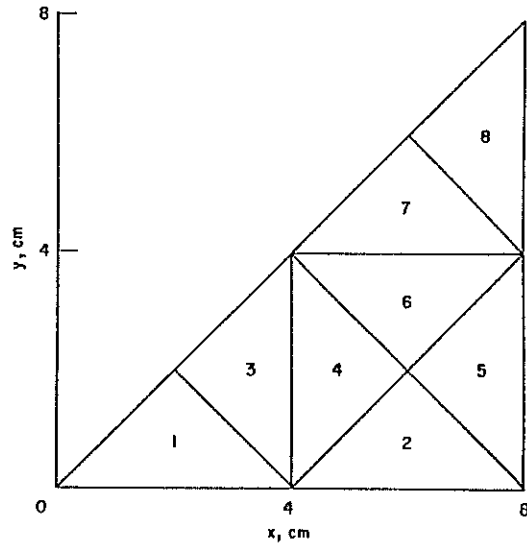


Fig. 2. Network on one eighth of a square.

The emf's which enter the loop equations involve the time derivatives of current and flux.

$$\sum_{k=1}^3 R_{jk} (I_{jk} + \Delta I_{jk}) = -\Delta \phi_{app,j} / \Delta t - \sum_{i=1}^{\ell} G_{ji} \Delta I_i / \Delta t \quad (4)$$

where R_{jk} and I_{jk} are respectively the resistance and current of side k of loop j , $\Delta \phi_{app,j}$ is the change in applied flux through loop j , and I and $I + \Delta I$ are the currents at time t and $t + \Delta t$ respectively. Equation (4) can be rewritten in matrix form

$$(G + R \Delta t) \Delta I = -\Delta \phi_{app} - (R \Delta t) I, \quad (5)$$

a system of m linear equations for the unknown ΔI . These equations plus the $n - 1$ node equations give the ΔI and thus the currents at time $t + \Delta t$. Fields, forces, and power can also be calculated at each time.

For details, especially with respect to the resistance of each line and concerning curved surfaces, see reference 4.

If one or more of the defined boundaries for a problem represent symmetry planes, the number of unknown currents and nodal equations are reduced. Lines which lie in the symmetry plane carry no currents, and nodes which lie in the symmetry plane obey Kirchhoff's node equation automatically. Thus the number ℓ of lines and the number n of nodes are chosen to exclude those in the symmetry planes. Consequently

$$\ell = m + n; \quad (6)$$

the number of unknown currents equals the number of loops plus non-trivial nodes.

3. COMPUTER PROGRAM DESCRIPTION

Overview

Figure 3 shows the relationship between the computer programs forming the EDDYNET package. EDDYMESH generates the triangular mesh for EDDY-COEF which calculates the coefficient matrix G . INVERT inverts this matrix for MOVITIME which calculates the currents, fields, and power for each time step, Δt . The remaining programs produce the graphical output for the movies and for diagnostic purposes. These programs will be described in detail below.

The computing system at Argonne consists of a time-sharing computer, IBM 360/75, which shares magnetic disks with the batch computers, IBM 370/195 and IBM 3033. A Tektronix model 4012 Graphic Display Terminal is used with the time-sharing system to diagnose the geometry and preview the movie frames. The movies are made on a III model FR80 plotting unit.

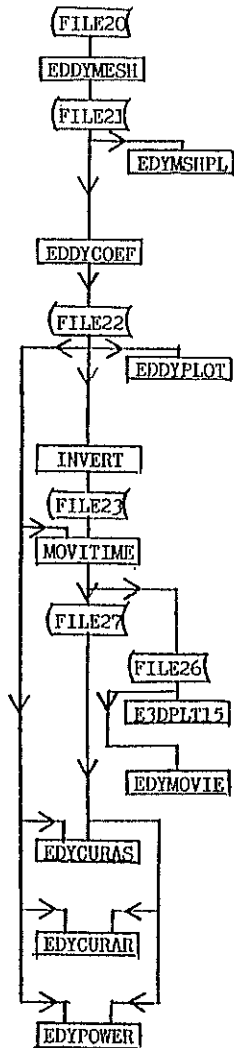
EDDYMESH

The triangle mesh generator of TRIM⁵ has been modified to facilitate setting up eddy current problems. The input data "cards" of file 20, listed in Fig. 4A, result in the mesh shown in Fig. 4B. The first card is the problem description card and the second card uses the namelist feature to specify the problem constants. Two additional problem constants are the thickness of the conductor and the time step, Δt , used in the program MOVITIME. The third card tells the program that for region 1 it should try to generate equilateral triangles of resistivity 1.0×10^{15} . Another option is to choose right triangles. The 6 cards that define the mesh and physical coordinates of the region follow. Similarly, region 2 is defined as having a resistivity of 6.0×10^{-6} and is overlayed on region 1. Hence, only 18 data cards are necessary to generate this mesh. The following symmetry codes may be specified:

- 1 - no symmetry
- 2 - symmetry about the y axis only
- 3 - symmetry about the x axis only

- 4 - symmetry about both x and y axis
- 8 - symmetry about the x axis and 45° line.

A symmetry code of 4 was used for this problem.



File 20 is the input data for EDDYMESH.

Calculates the triangular mesh and writes File 21 on disk. Is usually run in the time-sharing, TSO mode. Can be run in batch.

Eddy mesh plot. Plots the mesh on a Tektronix 4012 Graphic Display Terminal interactively on TSO.

Calculates the coefficient matrices G and GR and writes them in File 22. Can be run on TSO for up to an 8 x 8 mesh. Larger problems are run in batch.

Plots the mesh 4 times; first labeling the nodes, then the lines, then the loops, and finally the mesh points. Runs in TSO.

Inverts the G matrix and writes it in File 23. Can run the 8 x 8 mesh on TSO. Larger problems are run in batch.

Reads Files 22 and 23, and at each increment of time calculates the fields for a rectangular mesh and writes these in File 26. Calculates the currents for each line and writes them in File 27. The applied field shape is defined here.

Plots the magnetic field as a 3-D function of x and y for a 15 x 15 rectangular mesh.

Plots the magnetic field as colored triangles. Red is the high field, green is low field.

Reads Files 22 and 27 and plots the eddy currents as asterisks (*) on the triangle lines for each Δt of time.

Plots the eddy currents at each node as solid arrows whose area is proportional to the current.

Plots the instantaneous power at each node as an octagon whose area is proportional to the power.

Fig. 3. EDDYNET Computer Programs

PALTESE THICK CRSS. EDDYCURRENT TEST. 1/1E/78 R.J.LARI
 6PRGE PRBCON(1)=2.,7.,7.,0.,4., &ENC

		1.E+15	
6	1	1	0.0
	1	7	0.0
	4	7	4.5
	7	7	6.0
	7	4	6.0
	7	1	6.0
	2	1	6.E-06
6	1	1	0.0
	1	7	0.0
	4	7	4.5
	4	4	2.5
	7	4	6.0
	7	1	6.0

Fig. 4A. Input Data for EDDYMESH.

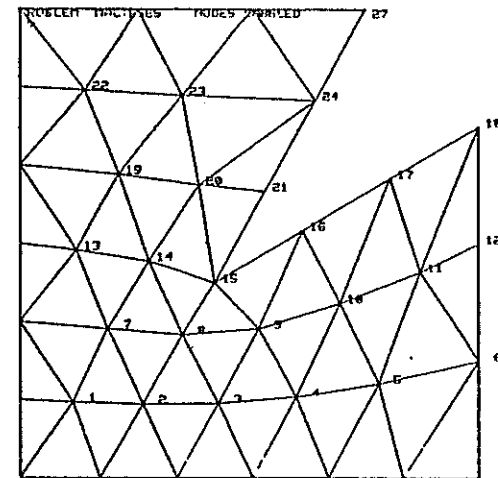


Fig. 4B. Mesh Generated by EDDYMESH.

By using a symmetry code of 8, only one region would be necessary to describe the problem. The input deck would then have a total of 8 data cards. This would also reduce the number of loops, lines, and nodes to approximately one half.

After the mesh is generated, the program also calculates the resistance of each triangle side. The results are then written on a magnetic disk as File 21.

EDYMSHPL

The mesh is generated using the time-sharing computer and the results may be immediately plotted on the Tektronix 4012 Graphic Display Terminal. The graphic subroutines of the Terminal Control System are used by this program. Any error in the input data can then be corrected and the mesh generated again.

EDDYCOEF

The G matrix of Eq. (5) is next calculated. Notice for Fig. 4B that there are 54 triangles (or loops), 40 nodes, and 93 lines. Symmetry considerations reduce these numbers to 27 non-trivial nodes and 81 lines. Those mesh points on the x and y axes are not treated as nodes because the corresponding node equations are satisfied trivially. For this case, the matrix G_{11} of Eq. (2) has 54 rows and 81 columns. The remaining 27 rows come from the node equations.

A program has been written that sweeps through the mesh tabulating the ends of each line and calculating the resistance of each line. It skips all mesh points if surrounded by air or if lying on a symmetry axis. A sweep through the lines locates and tabulates the six lines for each node. Finally, a sweep through the mesh tabulates the three lines for each loop and calculates the centroid of each loop. The G matrix is then calculated by sweeping through each line for each loop and is then stored on disk in File 22. The tabulations, resistances, and geometric information are also written in File 22.

For use in presenting the results of computation, a rectangular mesh is superimposed on the triangular mesh and a GR matrix for these mesh points is calculated. If a rectangular mesh point lies inside a loop, the centroid coordinates are used to calculate the field. The GR matrix is needed in a later program (MOVITIME) for calculating the fields on a rectangular mesh so that the field can be plotted as a three-dimensional surface by program E3DPLT15.

EDDYPLOT

This is a diagnostic program to check the output of EDDYCOEF on the Tektronix terminal. It enables one to plot the mesh showing the numbered nodes as in Fig. 4B, the numbered lines, the numbered loops, and also the numbered mesh points on separate plots. The graphic subroutines of the Tektronix Terminal Command System are used by this program.

INVERT

A standard matrix inversion program is used to invert the G matrix and write it in File 23.

MOVITIME

The time development of the eddy currents is calculated by this program using Eq. (5). For each of the time steps Δt , the program calculates the current in each line and the magnetic field at each

rectangular mesh point, these are written in Files 27 and 26 respectively, and are also printed for the first few time steps. In addition, the time and instantaneous power are also printed. Any function for the applied field can be specified in the program.

E3DPLT15

A 15 x 15 matrix of field values as read from File 26 is plotted by the program as a three-dimensional surface in Fig. 5. The graphic subroutines used by this program are from the DISSPLA⁶ proprietary software package. A card input specifies how many frames to plot and also chooses a linear or log B plot and the scale. The data in File 26 may be preplotted on the Tektronix 4012 by using a similar program in TSO. Selected frames are viewed and hard copies can be made. If desired, the viewpoint can easily be changed in the program. A variation of this program plots all four quadrants of the problem from the solution of the first quadrant as shown in Fig. 6.

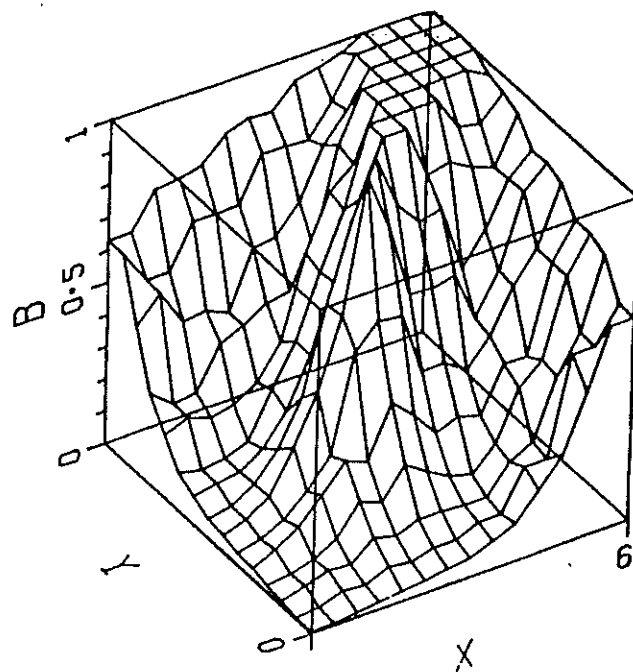


Fig. 5. A Three-Dimensional Field Plot of E3DPLT15.

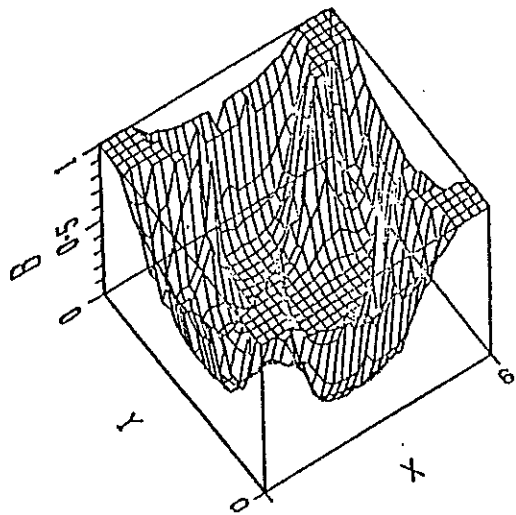


Fig. 6. Four Quadrants Plotted From First Quadrant Solution.

EDYMOVIE

This program colors the mesh triangles with the colors changing from green to red as the magnetic field increases. In the green-red color plane, the path chosen to change the color from the green axis to the red axis can be selected as: (1) a straight line joining the end of the red axis to the end of the green axis, (2) a circle through these two points, or (3) an equilateral hyperbola passing close to the yellow corner of the plane. Other options are to color the triangles from red to yellow or from green to yellow as the field increases. It is also possible to use a linear or log scale for the field. The DISSPLA⁶ graphic subroutines are used by this and the three following programs.

EDYCURAS

Figure 7 shows the output of this program for the first few time steps after the magnetic field has been applied. The motion of the asterisk along the line represents the motion of charges in the material along selected paths. The ratio of the current in the line to the maximum current in any line at any time determines how far the asterisk moves $1/3$ the length of the line for maximum current. If the distance exceeds the end of the line, it is returned to the start of the line and placed the remaining distance from the start. This program best utilizes the features of movies. A variation of this program plots all four $\pi/2$ sectors from a $\pi/2$ solution of a problem as shown in Fig. 8. A TSO version also exists.

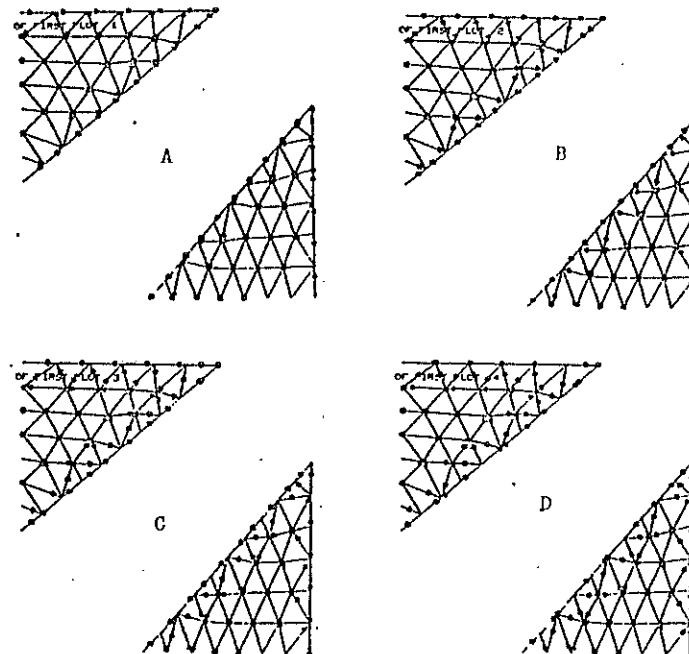


Fig. 7A-7D. First Four Time Steps Showing Asterisk Motion.

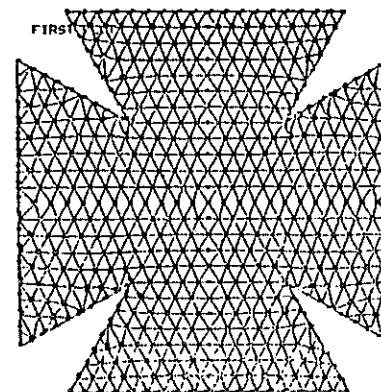



Fig. 8. Four Quadrants Plotted From One Quadrant Solution.

EDYCURAR

Another way of presenting the magnitude and direction of the eddy currents is by means of a solid arrow, e. g. . The area inside the arrow is proportional to the magnitude of the currents. This program reads the currents and other data from Files 27 and 22 respectively. The vector sum of all currents at each node is then calculated and plotted. The area of the arrows is normalized to a fixed maximum value which can be the same for each time step. Typical output is shown in Fig. 9. Selected time steps can be viewed using the TSO version of the program.

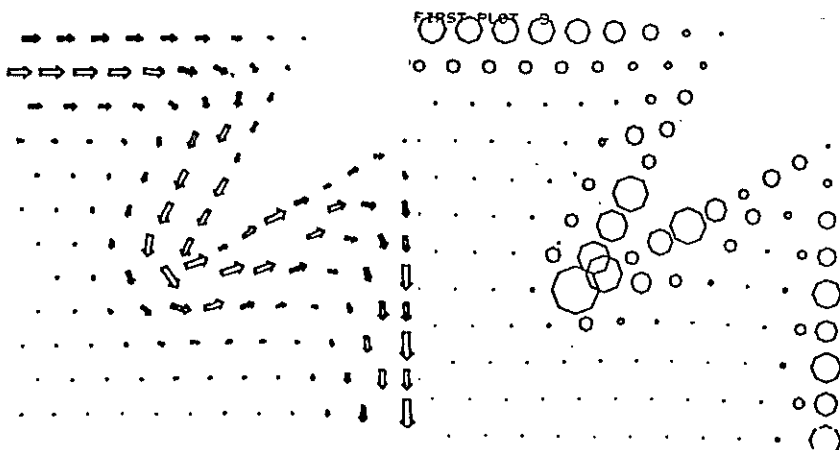


Fig. 9. Eddy Current Direction and Magnitude as Output of EDYCURAR

Fig. 10. Instantaneous Power Plotted by EDYPOWER.

EDYPOWER

The instantaneous power at each node is calculated by summing half the power of each line connected to the node. This is represented by the area of an octagon as shown in Fig. 10. The program reads the necessary data from Files 22 and 27. The areas of the octagons can be normalized to the maximum value for each time step. A TSO version allows any time step to be previewed.

Computer CPU Time and Size Requirements

Table 1 shows the program requirements for TSO and BATCH computations for two mesh sizes. The 7 x 7 mesh contains 27 nodes, 54 loops, and 81 lines. The 13 x 13 mesh has 128 nodes, 245 loops, and 373 lines. The computer time given for the last 6 programs is for 1000 time steps, Δt . The long time to color the triangles in EDYMOVIE makes this method undesirable as a method of presenting the field penetration. The three-dimensional field plots in black and white use one-tenth the time.

TABLE I. TIME AND SIZE REQUIREMENTS

Computer Program	Size (KBYTES)	Time (sec.)		Problem Size	
		TSO	Batch	7 x 7	13 x 13
EDDYMESH	78	~120		X	
EDDYCOEF	966 304	~60	59	X	X
INVERT	1164 266	2	443	X	X
MOVITIME	978 332	71	676	X	X
E3DPLT15	194 194	--	236	X	X
EDYMOVIE	164		2503	X	
EDYCURAS	152 152		763 197	X	X
EDYCURAR	198		414		X
EDYPOWER	192		433		X

4. GRAPHICAL PRESENTATION OF RESULTS

Fields

When one examines the printed output of current, field, and power for many values of time, it soon becomes apparent that a graphical presentation is desirable. A simple graphical presentation of the current direction is shown in Fig. 11. However, this does not indicate the magnitude of the current. Presenting the negative exponent of the magnetic field for each loop as shown in Fig. 11 is also useful to show the field penetration. A more useful presentation is shown in Fig. 5. Combining many of these instantaneous views in a movie results in a very clear understanding of how the field penetrates the conductor with time. The 16 mm black and white movies are exposed on the FR80 and the film developed once a day on a fixed schedule. Hence the "turn around" is not too long.

Currents

The eddy currents are visualized as a motion of "charges" in the conductor as a function of time. This can only be accomplished by making a movie, with each frame showing a position of the charges at an instant of time. Another way of representing the currents is by means of the solid arrows with area proportional to the magnitude of the current. This presentation can be useful either for viewing specific instants of time or by combining them in a movie.

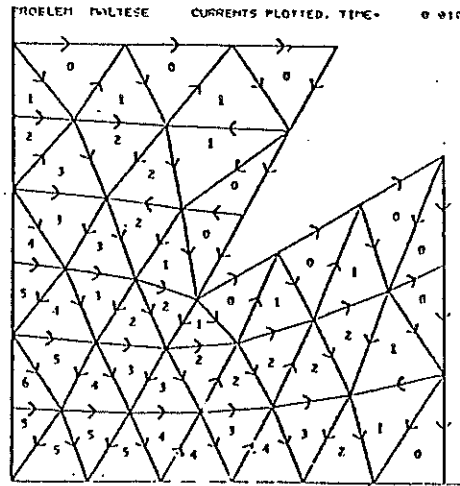


Fig. 11. The Current Direction and Field Magnitude Shown.

Power

The instantaneous power at a node, represented as the area of an octagon, can also be viewed for a single instant or in a movie. The power in each line could also be represented by the area of a rectangle placed on the line.

Quantitative Comparison

A quantitative comparison can be made between the results of MOVITIME and analytic calculations for some simple geometries. The power as a function of time, following the instantly applied magnetic field, has been fitted to the expression:

$$P = \sum_n A_n \exp(-2t/t_n).$$

A_n and t_n are then compared with the analytical values. For a circular cylinder with 49 mesh points per quadrant, the comparison is shown in Table II. The agreement is encouraging.

TABLE II

Fit to power for a circular cylinder: parameters A_n and t_n

	EDDYNET	Analytical		EDDYNET	Analytical
T_1 (s)	1.432	1.304	A_1 (MW)	11.79	11.94
T_2 (s)	0.283	0.247	A_2 (MW)	11.95	11.94
T_3 (s)	0.103	0.101	A_3 (MW)	17.42	11.94

Movie Content

A 15-min. movie has been prepared to illustrate the graphical output. Fourteen short sections of film of length 600 to 1000 frames show the following geometries:

Long Square Prism. Penetration of a step field into a long conducting prism of square cross section is represented as a three-dimensional surface. One quadrant is shown as in Fig. 5. The field is represented by color also. Low field is represented by green, high field by red, and intermediate fields by yellow and orange.

Long Maltese Cross Prism. A step field penetrating the prism is represented as a three-dimensional surface. The eddy currents are shown as a motion of dots in a full cross section and 1/8 cross section. The eddy currents are also shown as solid arrows.

Next, a sinusoidal field is shown penetrating one quadrant for a few cycles. The motion of dots to represent the eddy currents clearly shows the phase relationship of the currents. The solid arrows over one quadrant also show this. The peak instantaneous power is so large that the octagon representing it appears to jump out of the picture.

Thin Maltese Cross. A step field shows very large fields in the quadrilateral region between neighboring areas of the cross, due to the large eddy currents where the arms meet.

Long Hollow Cylinder. A sinusoidal field is applied for a few cycles and the resulting lack of field penetration into the hollow center illustrates good shielding properties.

5. FUTURE PLANS

The EDDYNET programs have been applied to only a limited class of problems. Several extensions of the programs are underway.

Curved Plates

Problems have been solved for curved surfaces approximated by nets of triangles. These are mentioned in reference 4, but they have not yet been incorporated into the version described above.

Three-Dimensional Solids

Three-dimensional solids can be modeled by tetrahedral elements. Loop and node equations similar to those described above can be written. The number of loop equations is diminished by the number of tetrahedra, since by Gauss' Law the net flux, and thus the net change of flux, out of each tetrahedron is zero.

Other future plans are to represent and graph current density throughout the conductor and to apply the EDDYNET technique to an array of long conductors joined at the ends in a specified way. Such an array has one-dimensional currents and two-dimensional fields, whereas EDDYNET deals with two-dimensional currents and one-dimensional fields. Most important, we hope to apply EDDYNET to the design of actual systems.

REFERENCES

1. C. J. Carpenter, "Computation of Magnetic Fields and Eddy Currents," Fifth Int. Conf. on Magnet Technology, Rome (1975), p. 147-158.
2. M. J. Newman, C. W. Trowbridge, L. R. Turner, "GFUN: An Interactive Program as an Aid to Magnet Design," Fourth Int. Conf. on Magnet Technology, Brookhaven (1972), p. 617-626.
3. C. S. Biddlecomb, C. J. Collie, J. Simpkin, C. W. Trowbridge, "The Integral Equation Method Applied to Eddy Currents," COMPUMAG Conference on the Computation of Magnetic Fields, Oxford (1976), p. 363-371.
4. L. R. Turner, "An Integral Equation Approach to Eddy-Current Calculations," IEEE Trans. Magn., MAG-13 1119 (1978).
5. J. S. Colonias, J. H. Dorst, "Magnet Design Applications of the Magnetostatic Program Called TRIM," Proceedings of the International Symposium on Magnet Technology, Sept. 8-10, 1965, SLAC Stanford, California.
6. DISSPLA, A proprietary software package of the Integrated Software Systems Corporation, P.O. Box 9906, San Diego, California 92109.

INTERACTIVE COMPUTER TECHNIQUES IN THREE DIMENSIONAL MODELISATION OF FIELD PROBLEMS BY FINITE ELEMENT METHOD

RAFINEJAD P., COULOMB J.L.

Laboratoire d'Electrotechnique de GRENOBLE . E.R.A. 534

BP. 46 38402 SAINT MARTIN D'HERES , FRANCE

ABSTRACT

The three dimensional input/output problems are directly approached through the basic notion of isoparametric finite element method. The latter is regarded as a numerical modelisation of space functions. The shape and the physical behaviour of the materials are projected, in the same manner, into the finite element topological discretized space. The resulting algorithms for the input/output interactions are discussed through the exemples of perspective and equipotential plots.

I - INTRODUCTION

L'emploi des méthodes numériques dans le calcul des champs magnétiques tridimensionnels est maintenant chose possible (réf. 1.2). Leur utilisation est cependant difficile à cause des problèmes de communication homme-machine posés par la modélisation des structures tridimensionnelles. Dans cette communication on distingue trois composantes : l'homme, l'ordinateur et le sujet. Ce dernier est un

problème physique caractérisé par un modèle mathématique. Cependant, la résolution analytique étant le plus souvent impraticable, on fait appel à des techniques numériques qui permettent de réaliser un modèle numérique potentiel à l'aide de l'ordinateur. L'homme suivant un certain nombre d'observations, de mesures et d'inventions, fournit des informations caractéristiques de l'application à l'ordinateur qui complète, à travers les techniques numériques appliquées aux lois physiques, le modèle numérique du sujet qui en retour sera étudié par l'homme (fig. 1).

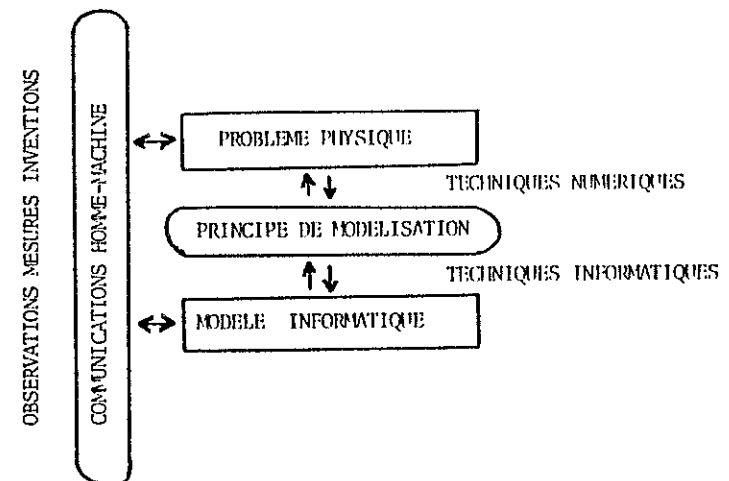


Fig. 1 Principe de la modélisation numérique.

Dans cet article, nous essayons d'approcher directement l'objectif final de la modélisation numérique, grâce au principe de la méthode des éléments finis.

II - MODÉLISATION PAR ÉLÉMENTS FINIS

Dans les problèmes de champ magnétique, l'objet est caractérisé par les formes des matériaux et les fonctions d'espace qui

définissent leurs comportements physiques. Dans cette approche, ces deux caractéristiques sont traitées, de la même manière, à travers des fonctions d'espace : les coordonnées x , y et z et le potentiel magnétique Ψ . La modélisation numérique consiste en une projection de ces fonctions dans un espace discrétisé d'éléments finis topologiques. Celui-ci est défini par un ensemble de noeuds et de liaisons qui forment des éléments topologiques à une, deux ou trois dimensions. La continuité de ce modèle est assurée par des polynômes d'interpolation normalisés et définis de manière unique. Pour toute fonction ψ nous retenons la formule d'interpolation :

$$\psi = \sum_{i=1}^n \alpha_i^1(u^1) \alpha_i^2(u^2) \alpha_i^3(u^3) \psi_i = \sum_{i=1}^n \alpha_i(u) \psi_i \quad (1)$$

où u^1 , u^2 et u^3 sont les coordonnées locales comprises entre -1 et +1 relatives à chaque élément topologique reliant n noeuds.

Cette interpolation appliquée aux coordonnées globales donne le modèle de la forme et définit en même temps les fonctions de transformation de coordonnées ce qui établit les modèles des opérateurs mathématiques (réf. 3). Ainsi à l'intérieur d'un élément l'opérateur gradient :

$$\vec{\nabla} \text{ est modélisé par } \sum_{i=1}^n \vec{C}_i \quad (2)$$

$$\text{avec } \vec{C}_i = g^{-1} \vec{\nabla} \alpha_i$$

où g est la matrice jacobienne de la transformation de coordonnées.

$$g_{kl} = \frac{\partial X^l}{\partial u^k} \quad k = 1,2,3 ; \quad l = 1,2,3 \quad (3)$$

et $\vec{\nabla}$ l'opérateur gradient dans l'espace normalisé.

Enfin l'opérateur d'intégration

$$\int dV \text{ est modélisé par } \sum_{l m n} \sum_{l m n} /g/ w w w \quad (4)$$

où w_1, w_m, w_n sont les poids d'une intégration numérique de GAUSS.

Le modèle physique, défini par la fonction de potentiel Ψ , sera déterminé à partir d'une formulation variationnelle (réf. 3).

Nous abordons le problème d'interaction graphique dans la modélisation géométrique ainsi que dans l'étude du comportement physique du modèle complet.

III - ENTRÉE DES DONNÉES GÉOMÉTRIQUES - MODÉLISATION DES FORMES

Cette étape consiste à transmettre les coordonnées réelles d'un certain nombre de points au moyen d'un modèle topologique. L'introduction des coordonnées des noeuds peut se réaliser soit par affectation directe, soit de manière interactive par identification avec un point appartenant à un modèle de l'objet. Un tel modèle pourra être obtenu soit par des techniques de dessin industriel soit par un assemblage, à la manière d'un jeu de construction, de volumes aux formes simples (réf. 4).

La figure 2 montre le cas d'un quart de cylindre avec son maillage topologique composé de 3 éléments cubiques à 27 noeuds et le maillage réel correspondant. Seule la connaissance de la position des noeuds clés (marqués d'une croix) est nécessaire car les coordonnées des autres noeuds peuvent se calculer par une série d'interpolations (réf. 3).

La figure 3 montre le maillage topologique et le maillage réel d'un dispositif à aimant permanent.

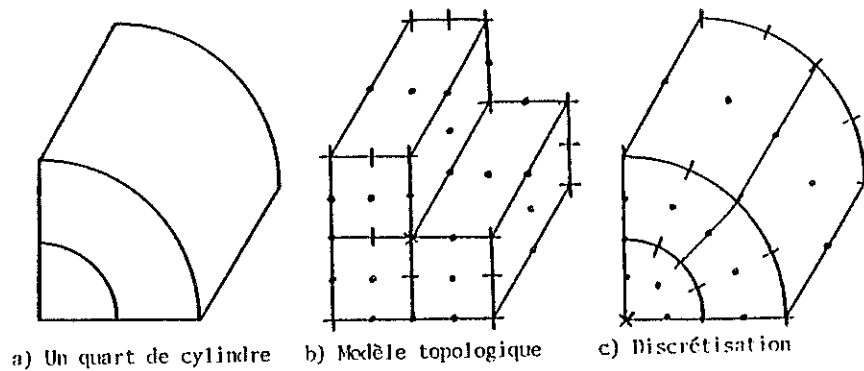


Fig. 2 Modélisation d'un quart de cylindre.

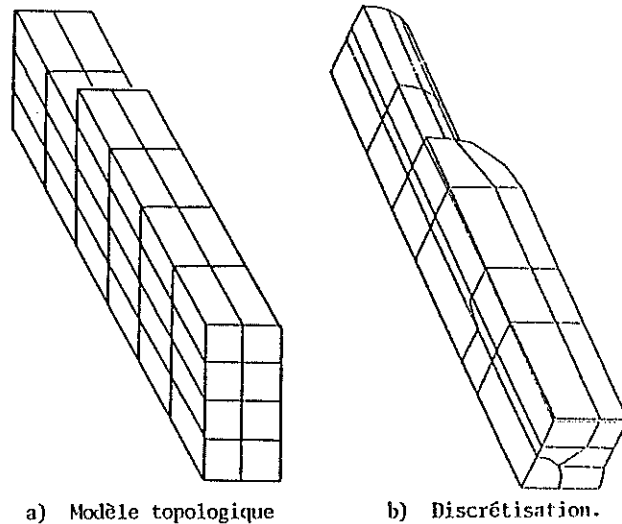


Fig. 3 Modélisation d'un dispositif à aimant permanent.

IV - INTERACTION GRAPHIQUE

Les informations concernant la géométrie de l'objet doivent être minutieusement contrôlée. Par ailleurs, il est nécessaire d'associer à certains sous-ensembles de la frontière du domaine les conditions aux limites. Enfin, les résultats obtenus par la méthode de résolution numérique doivent être appréhendés. Dans toutes ces opérations, l'interaction graphique est d'un grand secours. Le principe de la modélisation directe de la géométrie par la méthode des éléments finis conduit à des algorithmes simples et rapides de manipulation des surfaces et des volumes. Dans ces algorithmes, les fonctions logiques et les opérations entre entiers remplacent les formules algébriques complexes.

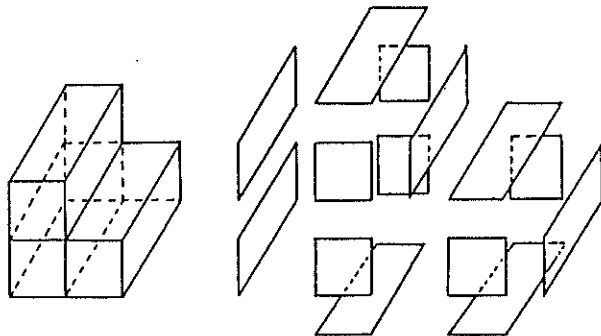
Nous dégageons les caractéristiques essentielles d'une telle interaction. Premièrement, il s'agit de visualiser sous différents points de vue, tout ou partie d'un volume ou d'une surface. Pour une meilleure compréhension des images, les projections se feront avec élimination des lignes cachées. Deuxièmement, la rapidité de production de ces images constitue un facteur déterminant dans l'efficacité de l'interaction. Finalement, ces images doivent conserver les liaisons avec le modèle d'origine. Ceci permet à l'utilisateur de désigner sur l'image une partie de l'objet, pour transmettre des informations.

Nous essayons d'atteindre ces objectifs suivant le processus ci-dessous :

- 1) Définition des surfaces susceptibles d'être vues (enveloppe).
- 2) Définition d'un point de vue par rotation de l'objet.
- 3) Elimination des lignes cachées à partir d'un point de vue.
- 4) Définition et manipulation de surfaces.

Détermination de l'enveloppe

Nous devons visualiser le modèle géométrique d'un objet, c'est-à-dire un ensemble d'éléments finis droits ou curvilignes assemblés de façon à approcher au mieux les formes de l'objet. Dans cet ensemble, seuls les éléments frontaliers sont susceptibles d'apparaître sur une image. En fait, seules les faces extérieures de ces éléments interviennent effectivement dans la définition de l'enveloppe. Ces facettes se distinguent des facettes intérieures par le fait qu'elles n'appartiennent qu'à un seul élément. Dans l'espace topologique où éléments de volume et de surface (les facettes) sont définis par des combinaisons de nombres entiers, les tests d'appartenance s'exécutent par l'intermédiaire de comparaisons simples et rapides (fig. 4).



a) Modèle topologique complet. b) Facettes de l'enveloppe.

Fig. 4 Détermination topologique de l'enveloppe du modèle.

Élimination des lignes cachées

Dans cette partie nous décrivons l'algorithme d'élimination des lignes cachées retenu pour la visualisation des structures tridimensionnelles modélisées par éléments finis. A ce stade, l'enveloppe de l'objet se compose d'un ensemble de facettes planes ou curvilignes que nous subdivisons, pour les besoins de l'algorithme, en sous facettes triangulaires en nombre plus ou moins grand suivant la finesse du tracé désiré (fig. 5).

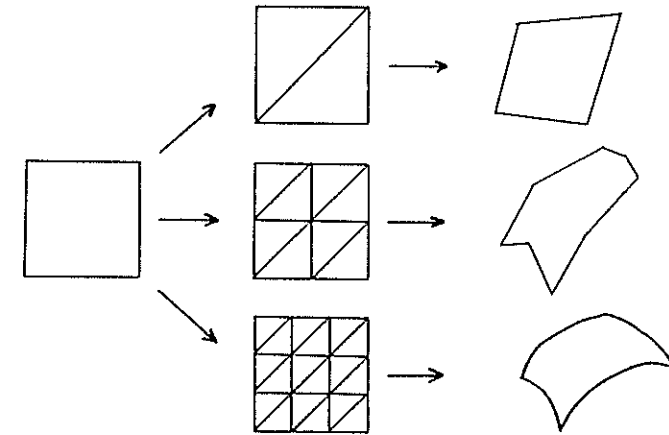


Fig. 5 Subdivision des facettes en triangles.

La visualisation consiste à tracer les arêtes des facettes ou éventuellement des sous-facettes afin de faire ressortir les courbures de l'enveloppe.

Le problème est alors le suivant : pour toutes les arêtes de l'enveloppe nous devons déterminer les segments non cachés par l'ensemble des triangles supposés opaques. C'est un problème de détermination de la position relative de ces éléments dans l'espace. Cet algorithme consiste à déterminer pour chaque triangle les segments d'arêtes visibles. La position relative des autres triangles est appréciée à partir de celle de l'ensemble des sommets, elle-même définie à partir des coordonnées d'aires. Nous procédons d'abord par l'élimination des faces qui n'ont aucune influence sur la visibilité des arêtes du triangle en question. Ceci réduit considérablement le nombre de combinaisons.

Nous déterminons la position de tous les points (X,Y,Z) de l'espace grâce à l'utilisation des coordonnées locales du triangle, suivant les formules :

$$\begin{pmatrix} l_1 \\ l_2 \\ l_3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ X_1 & X_2 & X_3 \\ Y_1 & Y_2 & Y_3 \end{pmatrix}^{-1} \cdot \begin{pmatrix} 1 \\ X \\ Y \end{pmatrix}$$

$$l_4 = Z - Z_1 l_1 - Z_2 l_2 - Z_3 l_3 \quad (5)$$

où X_i, Y_i et Z_i sont les coordonnées du ième sommet du triangle.

Autour du triangle nous définissons 14 sous-espaces (fig.6) auxquels nous associons un code suivant les relations :

$$N = 1_1 + 41_2 + 4^2 1_3 + 4^3 1_4 \quad (6)$$

où $1_i = 1$ si $l_i > 0$
 $1_i = 0$ si $l_i < 0$

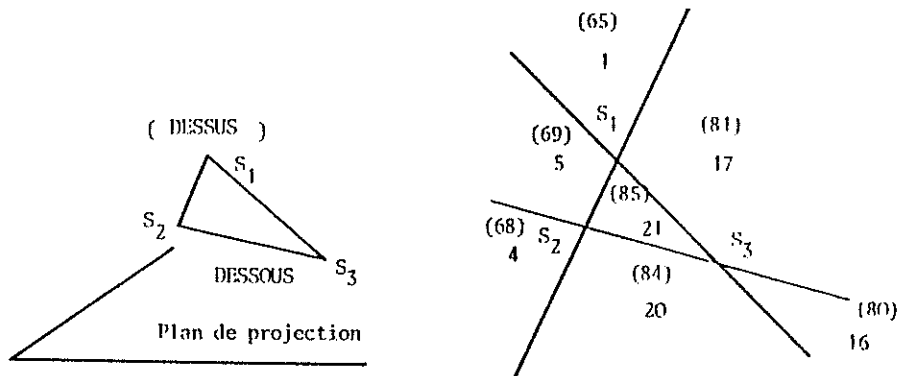


Fig. 6 Division de l'espace en 14 sous-espaces et codage.

Ce codage entraîne celui des triangles qui est utilisé pour le processus d'élimination. Ainsi tout triangle $S_4 S_5 S_6$ situé au-dessous du triangle est caractérisé par un code.

$$N_{456} = N_4 + N_5 + N_6 < 4^3 \quad (7)$$

De même, un triangle dont le code est tel que :

$$\text{MODULO}_{4^i} (N_{456}) < 4^{i-1} \quad (8)$$

avec $i = 1$ ou 2 ou 3

ne peut chevaucher le triangle (fig. 7).

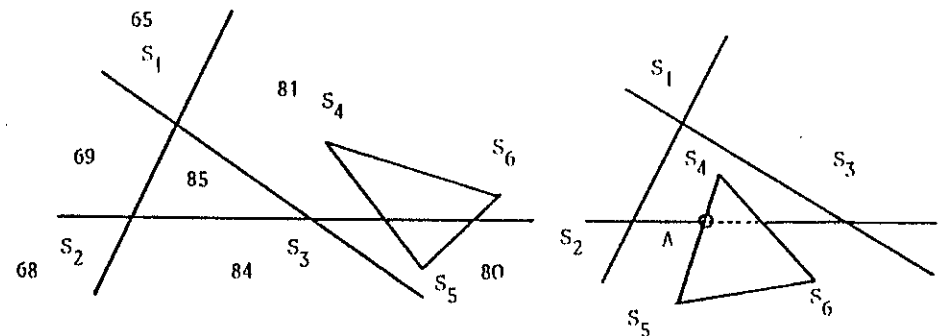


Fig. 7 Triangles disjoints.

Fig. 8 Intersection dans le plan de projection.

Les triangles qui subsistent après cette élimination topologique sont susceptibles de cacher une partie du triangle. Les intersections nécessaires à la détermination précise des parties cachées sont calculées algébriquement à partir des coordonnées d'aires (fig. 8), suivant les formules :

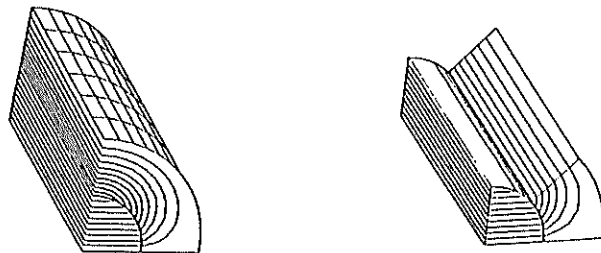
$$\begin{aligned}
 L_1^A &= 0 \\
 L_2^A &= \frac{L_1^5 L_2^4 - L_1^4 L_2^5}{L_1^5 - L_1^4} \\
 L_3^A &= 1 - L_1^A - L_2^A
 \end{aligned}
 \quad (9)$$

La figure 3b montre une application de l'algorithme. Le modèle est composé de 44 éléments finis curvilignes hexaédriques à 27 noeuds. L'enveloppe est formée de 80 facettes décomposées en près de 200 triangles. Ce tracé a nécessité environ 10 secondes de temps de calcul sur un IBM 360/67.

Manipulation de surfaces

Dans le processus de visualisation, les éléments de volume éclatent en plusieurs éléments de surface qui eux-mêmes se subdivisent en un certain nombre de triangles. Cependant, chaque nouvelle facette créée conserve la trace de ces origines. Cette mémorisation permet à l'utilisateur d'indiquer une face d'un élément en désignant sur l'image un des triangles qui la composent. Par des désignations successives, il peut associer plusieurs faces afin de composer une surface.

Nous utilisons cette facilité au cours de la définition interactive des conditions aux limites ainsi que pour la visualisation des résultats par tracé de courbe de niveau (fig. 9).



a) Vue des surfaces extérieures. b) Vue de surfaces intérieures.

Fig. 9 Courbes de niveau d'une fonction potentiel scalaire.

Par cette procédure en choisissant les éléments à visualiser nous avons accès à toute surface intérieure.

V - CONCLUSION

Nous avons abordé une partie de la modélisation tridimensionnelle, notamment l'interaction graphique adaptée à la modélisation par éléments finis. L'application de cette modélisation dépend en majeure partie de la performance des outils interactifs qui permettent à l'utilisateur d'introduire ses informations d'une manière naturelle et de mieux récupérer les renseignements utiles qui en résultent.

REFERENCES

1. MARROCCO A. Analyse numérique de problèmes Electrotechnique, Colloque IRA-CRM MONTREAL, Mai 1976.
2. RAFINEJAD P. Adaptation de la méthode des éléments finis à la modélisation des systèmes électromécaniques de conversion d'énergie. Thèse de Doctorat d'Etat ès-Sciences, INPG GRENOBLE, Janvier 1977.
3. RAFINEJAD P., COULOMB J.L., MEUNIER G. Permanent magnet three dimensional field computation by curvilinear finite elements. Paper 9.4 of this present proceeding of COMPUMAG 1978.
4. ANCELLE B. Definition and drawing of electrotechnical devices for field calculation. Paper 4.3 of this present proceeding of COMPUMAG 1978.