



**Institut Français de Recherche Scientifique
pour le Développement en Coopération**

INITIATION A LA NORME GRAPHIQUE GKS

Support de cours

Jean-Jacques LECHAUVE

Centre ORSTOM de Brest

BP 70 - 29280 PLOUZANE
Tél. 98 22 45 01

INITIATION A LA NORME GRAPHIQUE GKS

Support de cours

Jean-Jacques LECHAUVE

Centre ORSTOM de Brest

BP 70 - 29280 PLOUZANE
Tél. 98 22 45 01

Mars 1990

Remerciements

Ce travail a été réalisé au Centre ORSTOM de Brest sur Macintosh avec le logiciel Word. Je remercie tout particulièrement Michèle Joubert qui a largement contribué à la réalisation du document. Je désire également témoigner ma gratitude envers l'équipe du CICBx.

TABLE DES MATIERES**CHAPITRE 1 INTRODUCTION**

1.	Introduction	1
2.	Un brin d'histoire	2
3.	GKS à l'ORSTOM	4
4.	Introduction à la norme	5
4.1.	Pourquoi une norme en infographie	5
4.2.	Les exigences de GKS	5
4.3.	Objectif de conception	5
4.4.	Possibilités fonctionnelles	5
4.5.	Conception de l'interface utilisateur	6
4.6.	Appareils graphiques	6
4.7.	Mise en oeuvre	6
4.8.	Champ d'application	7
5.	Définitions	7
6.	Références bibliographiques	14

CHAPITRE 2 LE NOYAU GRAPHIQUE GKS

1. Introduction	15
2. Les sorties graphiques	16
2.1. Les primitives de sortie	16
2.2. Les attributs des primitives de sortie	16
2.2.1 Les attributs de polyligne	18
2.2.2 Les attributs de polymarque	18
2.2.3 Les attributs de texte	19
2.2.4 Les attributs de polygone	21
3. Les postes de travail	22
4. Les transformations	24
4.1. Coordonnées utilisateur et coordonnées universelles	24
4.2. Coordonnées normées d'appareil	24
4.3. Coordonnées d'appareil	24
4.4. Transformation de normalisation	25
4.5. Transformation de poste de travail	26
5. Les segments	28
5.1. Archivage des segments	28
5.2. Gestion des segments	28
5.3 Attributs des segments	29
6. Les entrées graphiques	29
7. Les métafichiers	30
8. Conclusion	31

CHAPITRE 3 DESCRIPTION DE CERTAINES PROCEDURES GKS

1. Des fonctions de contrôle	35
2. Les fonctions de transformation	39
3. Des fonctions de sortie et leurs attributs	42
4. Des fonctions de segments	47
5. Des fonctions de métachiers	53
6. Les fonctions d'entrée	55
7. Quelques fonctions d'interrogation	71

CHAPITRE 4 LES TRAVAUX PRATIQUES

1. Analyse de l'application	83
2. Exercice 1	86
3. Exercice 2	88
4. Exercice 3	91
5. Exercice 4	94
6. Exercice 5	96

ANNEXE A LISTE DES PROCEDURES GKS

Les fonctions de contrôle	99
Les fonctions de sortie	99
Les attributs de sortie	100
Les fonctions de transformation	101
Les fonctions de segmentation	101
Les fonctions d'entrée	102
Les fonctions de métachiers	103
Les fonctions d'interrogation	103
Les utilitaires	107
Les fonctions de gestion des erreurs	107

ANNEXE B LISTE DES ERREURS GKS

Les états	109
Les stations de travail	109
Les transformations	110
Les attributs de sortie	110
Les primitives de sortie	111
Les segments	111
Les appareils d'entrée	111
Les métachiers	112
Les fonctions d'échappement	112
Erreurs dépendant de l'implémentation	112

ANNEXE C LES TABLES DE GKS

Les tables de gks	113
-------------------	-----

ANNEXE D LE PACKAGE BBGKS

Les common de BBGKS	115
Le tracé des axes	116
Le tracé des tic-marques sur les axes	120
Le tracé des valeurs sur les axes	122
Positionnement des attributs de polyligne	126
Positionnement des attributs de polygone	127
Positionnement des attributs de polymarque	128
Positionnement des attributs de texte	129
Gestion des chaînes de caractères	130
Gestion/visualisation de boîte	132
Détermination d'une zone d'écho	134
Gestion des erreurs d'entrée/sortie	136

G K S

GRAPHIC

KERNEL

SYSTEM

1. INTRODUCTION

Le terme infographie (computer graphics) est apparu en 1974. Il provient d'un rapprochement des mots "informatique" et "graphique". En général, le mot infographie est qualifié par l'adjectif "interactive". Bien que souvent omis, il reste sous-entendu.

Le traitement d'image (image processing) doit être considéré comme une discipline distincte. L'infographie se définit donc comme l'utilisation de l'ordinateur pour manipuler des dessins. On peut également définir l'infographie comme l'utilisation d'objets graphiques pour représenter des données ou des résultats provenant de traitements informatiques.

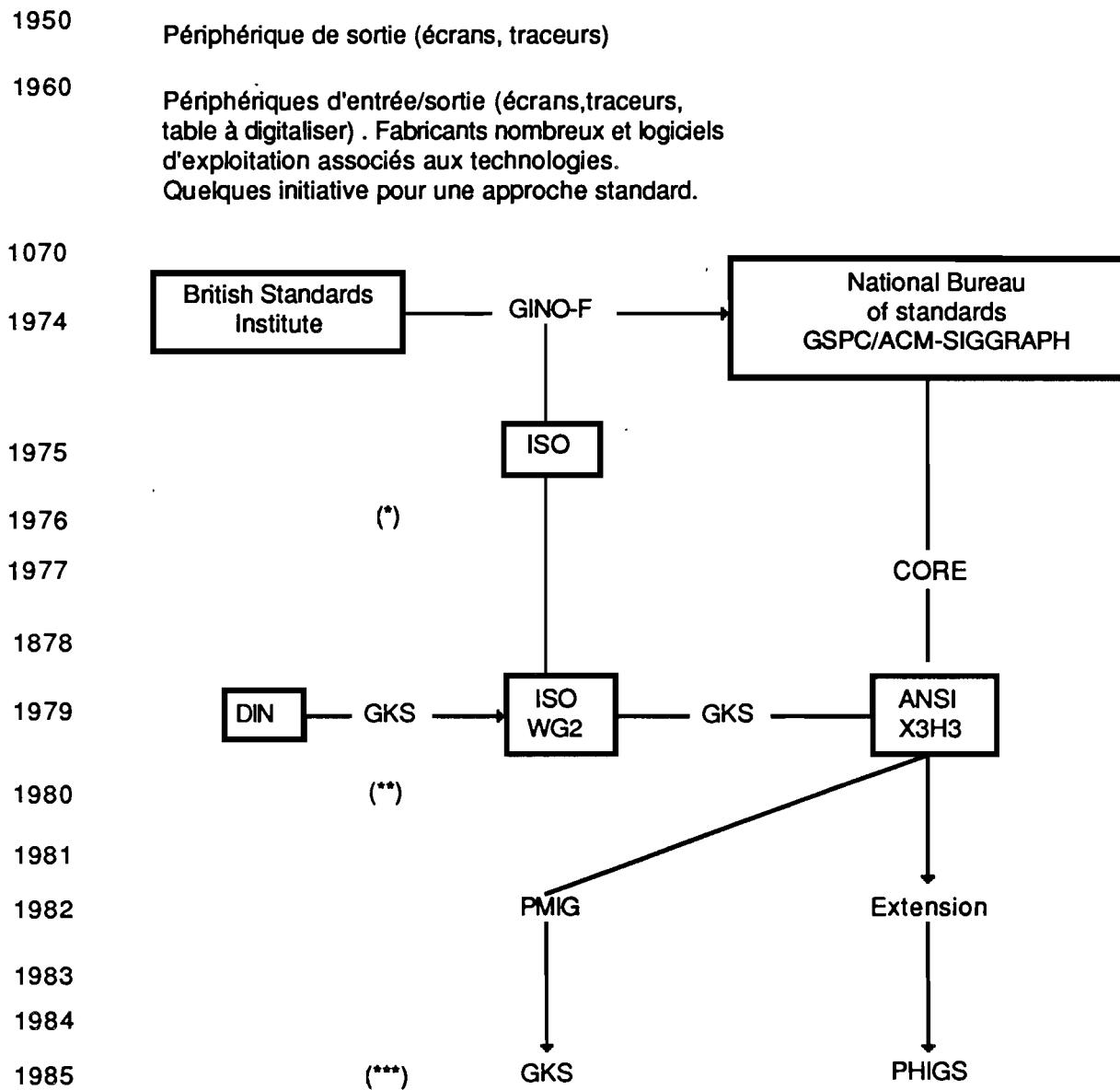
L'intérêt du graphique est reconnu depuis fort longtemps comme le rappelle le proverbe "une image vaut mille mots". L'apprehension d'un dessin est globale contrairement à la lecture d'un texte qui est une démarche séquentielle. Le dialogue homme-machine utilise de plus en plus fréquemment des icônes (petits symboles imaginés) qui sont mieux et plus vite perçus que des mots clés.

A ces débuts, l'informatique n'offrait que le traitement par lots (batch processing). Les terminaux de visualisation de moins en moins coûteux permettent d'introduire de plus en plus d'interactivité. Le domaine de l'infographie s'élargit pour devenir la conception assistée par ordinateur (CAO) (Computer Aided Design : CAD). Vers 1982, l'émergence des stations de travail graphiques conduit à l'épanouissement de la CAO. Les réseaux accroissent la puissance de ces stations plutôt dédiées à la restitution graphique ; les calculs qui deviennent de plus en plus volumineux restent à la charge des gros serveurs. La génération actuelle de station autorise un graphisme couleur à haute résolution. Des bibliothèques accessibles à partir des langages évolués offrent les outils de base (simples ou parfois, très sophistiqués) indispensables à la production d'application de qualité.

Les caractéristiques d'un système doivent être étudiées à trois niveaux différents : le matériel, le logiciel de base et le logiciel d'application. Pour le développement d'un logiciel, l'idéal est de disposer d'une bibliothèque graphique standard telle que GKS. S'investir dans l'emploi d'une norme, c'est s'offrir la liberté de choisir son matériel. Nous avons opté pour GKS-Bx (développé à l'Université de Bordeaux).

L'objectif de ce support de cours est essentiellement de présenter cette norme qui aujourd'hui fait l'objet d'une forte adhésion des développeurs de logiciels. À quelques restrictions près, GKS permet d'écrire des applications graphiques indépendantes des matériels. Les exercices proposés, présentent la mise en œuvre pratique sous FORTRAN de la norme, mais sont aussi conçus pour amener le développeur à analyser le dessin final en le décomposant en une suite d'objets graphiques élémentaires réutilisables.

2. UN BRIN D'HISTOIRE



(*) Séminaire de SEILLAC I (Château de Seillac, Vallée de la Loire). C'est l'origine de la standardisation en informatique graphique. L'une des décisions majeures fut à propos de la notion de transformation. La **construction d'un dessin et sa visualisation** doivent faire appel à des **transformations bien distinctes**.

(**) ANSI entreprend un effort de standardisation pour les couches basses.

VDI	Virtual Device Interface
VDM	Virtual Device Metafile

(***) VDI et VDM sont reconnus par ISO et deviennent CGI et CGM

ACM - SIGGRAPH	Association of Computing Machinery's Special Interest Group on Graphics
GSPC	Graphic Standard Planning Committee
DIN	Deutches Institut fur Normung
ISO	International Organization for Standardization
WG2	Working Group (ISO)
ANSI	American National Standard Institute
X3H3	Committee for computer graphics programming Languages (ANSI)
GINO-F	
CORE	<noyau graphique>
PMIG	Programmer's Minimal Interface to Graphics
GKS	Graphic Kernel System
PHIGS	Programmer's Hierarchical Interactive Graphics Standard
CGI	Computer Graphic Interface C'est la réponse aux nombreux constructeurs de périphériques graphiques qui désiraient un interface commun afin de permettre des liaisons rapides de leur équipements aux calculateurs
CGM	Computer Graphic Metafile Répond à la même préoccupation mais plutôt en terme de communication

3. GKS A L'ORSTOM

4. INTRODUCTION A LA NORME

Le système graphique GKS (Graphical Kernel System) est constitué d'un ensemble de fonctions destinées à la programmation en infographie. GKS est un système graphique de base susceptible d'être utilisé dans la majorité des applications produisant des images au moyen d'ordinateurs.

4.1. POURQUOI UNE NORME EN INFOGRAPHIE ?

- a) Portabilité : Pour permettre de transporter facilement des programmes d'application utilisant l'infographie d'une installation à une autre.
- b) Aide à la programmation graphique.
- c) Servir de cahier des charges : pour les constructeurs d'équipements graphiques en fournissant des combinaisons utiles de fonctions graphiques pour un appareil.

4.2. LES EXIGENCES DE GKS

- a) GKS doit inclure toutes les possibilités qui sont essentielles dans tout le spectre de l'infographie, des sorties passives aux applications interactives.
- b) Toute la gamme d'appareils graphiques, que ce soient les appareils à quadrillage ou à balayage cavalier, les enregistreurs de microfiches, les tubes à mémoire ou les consoles de visualisation à rafraîchissement, doit être contrôlée par GKS de la même façon.
- c) GKS ne doit pas être encombrant en place mémoire.

4.3. OBJECTIF DE CONCEPTION

- a) Cohérence : les exigences obligatoires de GKS ne doivent pas être contradictoires.
- b) Compatibilité : aucune autre norme ou règle couramment acceptée ne doit être violée.
- c) Orthogonalité : si les fonctions ou modules de GKS sont dépendants les uns des autres, la dépendance doit être bien définie et structurée, le cas normal étant l'indépendance.

4.4. POSSIBILITES FONCTIONNELLES

- a) Complétude : toutes les fonctions utiles à la majorité des applications d'un niveau de fonctionnalité donné seront incluses.
- b) Minimalité : les fonctions qui ne sont pas indispensables pour les applications d'un niveau de fonctionnalité donné ne seront pas incluses.
- c) Compacité : une application devra pouvoir atteindre le résultat désiré en utilisant un ensemble de fonctions et de paramètres aussi réduit que possible.

d) Richesse : un ensemble large de fonctions offre des facilités importantes allant au-delà des fonctions de base et comprenant des possibilités d'ordre supérieur.

4.5. CONCEPTION DE L'INTERFACE UTILISATEUR

- a) Ergonomie : GKS permettra la conception d'interfaces utilisateurs agréables.
- b) Clarté : les concepts et les possibilités fonctionnelles de GKS seront faciles à comprendre, en particulier par les programmeurs d'applications.
- c) Traitement des erreurs : les défaillances de modules ou de fonctions de système qu'elles soient provoquées par les erreurs du système lui-même ou par le programme d'application seront traitées de façon telle que la réaction à l'erreur soit suffisamment compréhensible et informative pour le programmeur d'application et que les conséquences sur le système et sur le programme d'application soient aussi réduites que possible. Le traitement des erreurs fait partie intégrante de GKS, cela pour aider à la clarté.
La spécification de GKS est divisée en une description générale, une description des structures de données sous-jacentes représentant l'état du système et une description des fonctions et de leurs effets sur ces structures de données.

4.6. APPAREILS GRAPHIQUES

- a) Indépendance vis-à-vis des appareils : les fonctions de GKS seront conçues de façon à permettre aux programmes d'application, utilisant ces fonctions, d'exploiter les facilités d'appareils graphiques d'entrée ou de sortie très différents sans modification de la structure de ces programmes.
- b) Richesse des appareils : toutes les possibilités d'une large gamme d'appareils graphiques d'entrée ou de sortie seront accessibles à partir des fonctions de GKS.

Tous ces principes ont conduit au concept fondamental sous-jacent de l'architecture de GKS : le concept de plusieurs postes de travail graphiques indépendants connectés à GKS et pilotés par ce système. Le programme d'application peut s'enquérir des possibilités de chaque poste de travail. Le système GKS comprend des fonctions d'échappement qui sont aisément identifiables dans un programme d'application et qui peuvent être utilisées pour accéder à des facilités particulières d'un appareil donné.

4.7. MISE EN OEUVRE

- a) Facilité de mise en oeuvre : les fonctions de GKS devront pouvoir être supportées par la plupart des langages hôtes, sur la plupart des systèmes d'exploitation, et avec la plupart des appareils graphiques.
 - b) Indépendance vis-à-vis du langage : il doit être possible d'accéder aux fonctions de GKS en utilisant l'un quelconque des langages de programmation normalisés par l'ISO.
 - c) Efficacité : la norme doit pouvoir être mise en oeuvre sans requérir l'utilisation d'algorithmes coûteux en temps.
 - d) Robustesse : l'opérateur et le programmeur d'application seront protégés le mieux possible vis-à-vis des défaillances de matériels ou de logiciels du système.
-

4.8. CHAMP D'APPLICATION DE LA NORME

L'introduction de cette norme devrait clarifier l'univers du logiciel graphique et favoriser considérablement ce marché tant sur le plan du graphique de gestion que des applications techniques et de la CAO.

Son principal avantage sera de permettre une portabilité des applications graphiques au niveau des codes sources dans les langages de haut niveau (Fortran, Pascal...) et leur indépendance vis-à-vis des périphériques.

GKS est un modèle conceptuel en même temps qu'une syntaxe, utilisé pour incorporer du graphisme dans un programme d'application.

GKS normalise six primitives graphiques et le mode de gestion des attributs qui leur sont rattachés, indépendamment de tout périphérique qui est "vu" par l'application comme "station de travail" logique.

La distinction entre un terminal graphique intelligent capable de générer en local certaines fonctions (remplissage de surfaces, marqueurs, génération de figures...) et une simple imprimante graphique ne sera pas effectuée à ce niveau.

Une console graphique et un traceur, par exemple, ne seront distingués qu'au niveau du contrôleur d'interface (driver). On voit donc bien se dessiner les deux pièces maîtresses de GKS c'est-à-dire la partie Virtuel qui est GKS lui-même et la partie Interface avec les différentes stations de travail disponibles. Cette seconde partie de GKS n'est, à l'heure actuelle, soumise à aucune norme.

5. DEFINITIONS

Ce chapitre donne la définition des notions utilisées dans le système GKS. Dans la mesure du possible, la terminologie utilisée est celle employée couramment en infographie.

ACCUSE DE RECEPTION (acknowledgment) :

Sortie indiquant à l'utilisateur d'un appareil logique d'entrée qu'une gachette a été activée.

APPAREILS DE VISUALISATION - DISPOSITIF GRAPHIQUE (display device - graphics device) :

Un dispositif (par exemple tube à rafraîchissement, tube à mémoire, traceur) sur lequel on peut afficher des images.

APPAREIL LOGIQUE D'ENTREE (logical input device) :

Une abstraction réalisée par un ou plusieurs dispositifs physiques qui rendent des valeurs d'entrée logiques au programme. Dans GKS, les appareils logiques d'entrée peuvent être de type : releveur de coordonnées, releveur d'une suite de coordonnées, entrée de scalaire, sélecteur, désignation et entrée de chaîne de caractères.

ASPECTS DES PRIMITIVES (aspects of primitives) :

Façons dont l'apparence d'une primitive peut varier. Certains aspects sont définis directement par des attributs de primitive, certains autres sont définis indirectement par une table de groupage. Un aspect, la mise en valeur, des primitives internes à un segment est commandé par le segment lui-même. Ceci ne s'applique pas aux primitives hors segment.

ATTRIBUTS (attribute) :

Une propriété particulière qui s'applique à un élément graphique (primitive de sortie) ou à un segment. Exemples : hauteur des caractères, mise en valeur. Notons que dans GKS, certaines caractéristiques des postes de travail sont appelées attributs de poste de travail.

ATTRIBUT DE PRIMITIVE (primitive attribute) :

Les valeurs des attributs de primitive (primitives de sortie) sont choisies par l'application de façon indépendante du poste de travail mais peuvent avoir des effets dépendants du poste de travail.

ATTRIBUTS DE SEGMENTS (segment attributes) :

Attributs qui ne s'appliquent qu'aux segments. Avec GKS, les attributs de segment sont la visibilité, la mise en valeur, la détectabilité, la priorité de segment et la transformation de segment.

CLASSE D'ENTREE (input class) :

Un ensemble d'appareils d'entrée logiquement équivalents au niveau de leur fonction. Dans GKS, les classes d'entrée sont : releveur de coordonnées, releveur d'une suite de coordonnées, entrée de scalaire, sélecteur, désignation et entrée de chaîne de caractères.

CLOTURE (viewport) :

Une partie de l'espace de coordonnées NDC définie par le programme d'application.
Note : dans GKS, cette définition est limitée à une région rectangulaire de l'espace des coordonnées NDC utilisée pour définir la transformation de normalisation.

CLOTURE DU POSTE DE TRAVAIL (workstation viewport) :

Une partie de l'espace de visualisation courant pour les sorties graphiques.

COORDONNEE D'APPAREIL (device coordinate) (DC) :

Coordonnée exprimée dans un système de coordonnées dépendant de l'appareil.
Note : dans GKS, l'unité de DC est le mètre sur un appareil capable de produire une image à une taille précise. Sinon des unités adaptées et dépendantes du poste de travail sont utilisées.

COORDONNEES NORMEES (normalized device coordinates) (NDC) :

Une Coordonnée exprimée dans un système intermédiaire indépendant de l'appareil et normée sur un certain domaine tel (0,1).
Note : dans GKS, pendant l'étape intermédiaire, les coordonnées peuvent être hors du domaine spécifié, mais l'information de découpage associée assure que la sortie n'excédera pas le domaine (0,1) x (0,1).

COORDONNEES UNIVERSELLES-UTILISATEUR (world coordinate) (WC) :

Un système de coordonnées cartésiennes indépendant et utilisé par le programme d'application pour spécifier les sorties et les entrées.

DECOUPAGE (clipping) :

Suppression des parties d'éléments graphiques situées à l'extérieur d'une limite, habituellement une fenêtre ou une clôture.

DISPOSITIF DE DESIGNATION (pick device) :

Appareil logique d'entrée qui retourne l'identificateur de désignation de la primitive désignée et le nom du segment associé.

DISPOSITIF D'ENTREE DE CHAINE DE CARACTERES (string device) :

Un appareil logique d'entrée de GKS fournissant en retour une chaîne de caractères.

ECHAPPEMENT (escape) :

Une fonction de GKS utilisée pour accéder à des facilités dépendantes de la réalisation ou de l'appareil autres que la génération de sortie graphique et inaccessibles autrement par GKS.

ECHO (echo) :

La notification immédiate à l'opérateur, par l'intermédiaire de la console de visualisation, de la valeur courante produite par un dispositif d'entrée.

ELEMENT GRAPHIQUE (display element) :

Un élément de base utilisé pour construire une image.

ENTREE DE SCALAIRES (valuator device) :

Un appareil d'entrée logique de GKS retournant un nombre réel.

ESPACE D'AFFICHAGE (display space) :

La portion de l'espace d'affichage correspondant à la surface disponible pour afficher des images. Dans GKS, le terme espace d'affichage est aussi utilisé pour désigner l'espace de travail d'un dispositif d'entrée de coordonnées.

ESPACE D'APPAREIL (device space) :

L'espace défini par les points adressables d'un dispositif de visualisation.

FENETRE (window) :

Une fenêtre prédéfinie d'un espace virtuel.

Note : dans GKS, cette définition est limitée à une région rectangulaire de l'espace WC utilisée pour définir la transformation de normalisation.

FENETRE DU POSTE DE TRAVAIL (workstation window) :

Une zone rectangulaire du système de coordonnées NDC projetée sur un espace de visualisation.

GACHETTE (trigger) :

Un dispositif d'entrée physique ou un ensemble de dispositifs qu'un opérateur peut utiliser pour indiquer des moments significatifs dans le temps.

GERANT D'APPAREIL (device driver) :

La partie dépendante de l'appareil, dans une réalisation de GKS, destinée à piloter l'appareil graphique. Le gérant d'appareil génère les sorties, prend en charge les interactions dépendantes de l'appareil.

GRAPHIQUES PAR QUADRILLAGE (raster graphics) :

Infographie où l'image est composée d'une trame de pixels disposés en lignes et colonnes.

IDENTIFICATEUR DE DESIGNATION (pick identifier) :

Un nom, lié à des primitives graphiques internes à un segment et retourné par le dispositif de désignation. Le même identificateur de désignation peut être attaché à plusieurs primitives graphiques.

IMAGE (display image ; picture) :

L'ensemble des éléments graphiques ou des segments affichés à un instant donné sur une surface de visualisation.

INDEX DE GROUPAGE (bundle index).

Un index dans une table de groupage associée à une primitive de sortie particulière. L'index définit les aspects de la primitive qui dépendent du poste de travail.

MARQUE (marker) :

Un symbole graphique d'allure spécifique utilisé pour repérer une position particulière.

MATRICE DE PIXEL (cell array) :

Une primitive de sortie de GKS consistant en une grille rectangulaire de cellules rectangulaires et égales, chacune ayant sa propre couleur.

Note : ces cellules ne sont pas forcément projetées de façon biunivoque sur les pixels de la surface de visualisation.

MESURE (measure) :

Une valeur (associée à un appareil logique d'entrée) déterminée par un ou plusieurs appareils physiques d'entrée et obtenue à partir des valeurs provenant des appareils physiques. La valeur d'entrée logique rendue par un appareil logique d'entrée est la valeur courante de cette mesure.

METAFICHIER GKS (GKS metafile) (GKSM) :

Un fichier séquentiel qui peut être écrit et lu par GKS : utilisé pour le stockage d'informations graphiques à long terme (également pour les transmettre et les transférer).

MISE A L'ECHELLE, ZOOM (scaling ; zooming) :

Agrandir ou réduire tout ou une partie de l'image en multipliant les coordonnées des éléments graphiques par une constante.

Note : pour des échelles différentes sur les deux directions orthogonales, on doit utiliser deux constantes différentes. Note : dans GKS, cette transformation n'est applicable qu'aux segments.

MISE EN VALEUR (highlighting) :

Une possibilité, indépendante des appareils, d'attirer l'attention sur un segment en modifiant ses attributs visuels (une généralisation du clignotement).

NIVEAU DE GKS (GKS level) :

Deux valeurs dans les domaines (0,2) et (a,c) qui, ensemble, définissent un sous-ensemble de possibilités fonctionnelles de GKS. Une mise en oeuvre de GKS fournit précisément toutes les fonctions d'un niveau.

NOTIFICATION (prompt) :

sortie indiquant à l'opérateur qu'un dispositif logique d'entrée donné est disponible.

OBLIGATOIRE POUR LES POSTES DE TRAVAIL (workstation mandatory) :

Dit d'une caractéristique qui doit être réalisée de manière identique sur tous les postes de travail d'une réalisation de la norme.

OPERATEUR, UTILISATEUR (operator) :

La personne qui manipule les moyens d'entrée physiques afin de changer leur "mesure" et qui actionne leurs gâchettes.

PIXEL : ELEMENT D'IMAGE (pixel ; picture element) :

Le plus petit élément d'une surface de visualisation auquel on puisse assigner une couleur et une intensité.

POLICE ET PRECISION DU TEXTE (font and precision) :

Un aspect du texte dans GKS, à deux composants, qui détermine la forme des caractères à afficher sur un poste de travail particulier. En plus, la précision décrit la fidélité de reproduction des autres attributs d'aspect du texte demandés par le programme d'application. Dans l'ordre de fidélité croissante, les précisions sont : chaîne, caractère et haute qualité.

POLYGONE (fill area) :

Une primitive graphique de GKS consistant en une frontière polygonale fermée qui peut être transparente ou remplie par une couleur uniforme, un motif répété ou des hachures d'un certain type.

POLYLIGNE (polyline):

Une primitive de GKS constituée par une suite de segments de droite connectés.

POLYMARQUE (polymarker) :

Une primitive graphique de GKS consistant en un ensemble de positions indiquées par une marque.

POSTE DE TRAVAIL (workstation) :

GKS est basé sur la notion de postes de travail abstraits, assurant l'interface logique par laquelle le programme d'application contrôle les appareils physiques.

PRIMITIVE DE SORTIE, PRIMITIVE GRAPHIQUE (output primitive, graphic primitive) :

Un élément graphique. Les primitives de sortie dans GKS sont : la polyligne, la polymarque, le texte, le polygone, la matrice de pixels et la primitive graphique généralisée.

PRIMITIVE GRAPHIQUE GENERALISEE (generalized drawing primitive) (GDP):

Une primitive graphique utilisée pour accéder à des facilités géométriques spéciales de poste de travail comme la génération de courbes.

PRIORITE DE SEGMENT (segment priority) :

Un attribut de segment utilisé pour savoir, lorsqu'ils se recouvrent, quels segments ont préséance pour la sortie graphique et l'entrée.

REALISATION OBLIGATOIRE (implementation mandatory) :

Une réalisation obligatoire décrit une propriété qui doit être réalisée de façon identique pour tous les postes de travail de toutes les réalisations de la norme.

RELEVEUR DE COORDONNEES (locator device) :

Un appareil d'entrée logique de GKS fournissant une position en coordonnées WC et un numéro de transformation de normalisation.

RELEVEUR DE SUITE DE COORDONNEES (stroke device) :

Un appareil logique d'entrée de GKS fournissant une suite de positions en coordonnées WC et un numéro de transformation de normalisation.

RETOUR (feedback) :

Sortie indiquant à l'opérateur l'interprétation d'une valeur d'entrée logique interprétée par le programme d'application.

ROTATION (rotation) :

Faire subir une rotation à tout ou une partie de l'image autour d'un axe.
Note : dans GKS, cette transformation n'est applicable qu'aux segments.

SEGMENT (segment) :

Ensemble de primitives graphiques manipulables comme un tout.

SELECTEUR (choice device) :

Un appareil logique d'entrée de GKS fournissant un entier non négatif représentant une sélection parmi un ensemble de choix.

SURFACE DE VISUALISATION, SURFACE D'AFFICHAGE (display surface, view surface) :

Sur un appareil d'affichage, le support sur lequel peuvent apparaître les images.

TABLE DE GROUPAGE (bundle table) :

Une table dépendante du poste de travail est associée à une primitive de sortie particulière. les entrées de la table donnent tous les aspects de primitive dépendants du poste de travail. Dans GKS, les tables de groupage existent pour les primitives de sortie suivantes : polyligne, polymarque, texte et polygone.

TABLE DE GROUPAGE DE POLYGONE (fill area bundle table) :

Une table associant à un index de polygones des valeurs spécifiques des attributs d'aspects de polygone dépendants du poste de travail. Dans GKS, cette table contient, pour chaque entrée, le style de remplissage, l'index de style et l'index de couleur.

TABLE DE GROUPAGE DE POLYLIGNE (polyline bundle table) :

Une table associant à un index de polyligne des valeurs spécifiques de tous les attributs de polyligne dépendants du poste de travail. Dans GKS, une entrée de cette table comprend le type de trait, le coefficient d'épaisseur de trait et l'index de couleur.

TABLE DE GROUPAGE DE POLYMARQUE (polymarker bundle table) :

Une table associant à un index de polymarque des valeurs spécifiques de tous les attributs d'aspect de polymarque dépendants du poste de travail. Dans GKS, une entrée de cette table comprend le type de marque, le coefficient de taille de marque et l'index de couleur.

TABLE DE GROUPAGE DE TEXTE (text bundle table) :

Une table associant à un index de texte des valeurs spécifiques de tous les attributs d'aspect de texte dépendants du poste de travail. Dans GKS, une entrée de cette table comprend la police et la précision du texte, le facteur d agrandissement des caractères, l'espacement inter-caractères et l'index de couleur.

TABLE DES COULEURS (colour table) :

Une table dépendant du poste de travail dont chaque entrée définit une couleur particulière à l'aide de valeurs des intensités de rouge, vert et bleu.

TEXTE (text) :

Une primitive graphique de GKS consistant en une chaîne de caractères.

TRANSFORMATION DE NORMALISATION - TRANSFORMATION DE VISUALISATION - TRANSFORMATION DE FENETRE A CLOTURE (normalisation transformation - viewing transformation - window-to-viewport transformation) :

Une transformation qui fait correspondre la limite et l'intérieur de la fenêtre à la limite et l'intérieur de la clôture.

Note : dans GKS, cette transformation fait correspondre des positions en coordonnées WC à des positions en coordonnées NDC.

TRANSFORMATION DE POSTE DE TRAVAIL (workstation transformation) :

Une transformation qui projette la limite et l'intérieur de la fenêtre du poste de travail sur la limite et l'intérieur de la clôture du poste de travail (partie de la surface d'affichage) en préservant les proportions en x et y.

Note : dans GKS, cette transformation projette des coordonnées NDC en coordonnées DC. Le résultat du respect des proportions en x et en y est que l'intérieur de la fenêtre du poste de travail peut ne pas recouvrir toute la clôture du poste de travail.

TRANSFORMATION DE SEGMENT (segment transformation) :

Une transformation qui est telle que les primitives du segment peuvent apparaître dans des positions (translation), tailles (échelle) et/ou orientations (rotation) variables sur la surface d'affichage.

TRANSLATION (translation : shift) :

L'application d'un déplacement constant aux positions de tout ou une partie de l'image affichée.
Note : dans GKS, cette transformation n'est applicable qu'aux segments.

VALEUR D'ENTREE LOGIQUE (logical input value) :

Une valeur rendue par un appareil logique d'entrée.

6. REFERENCES BIBLIOGRAPHIQUES

Infographie

Système graphique de base (GKS)
Description fonctionnelle
NF EN 27942, AFNOR, déc. 1986

Le système graphique GKS

F.R.A. HOPGOOD.
D.A. DUCE, J.R. GALLOP et D.C. SUTCLIFFE
Masson (ed.), 1986

Le système graphique GKS

Richard GOULETTE
Editests (ed.), 1986

GKS

Mise en oeuvre en FORTRAN
Joël LANGLA, Françoise MAZAT-GASTARRIET
Tecknea (ed.), 1988

GKS Bx

Versions xxx et 7.1.
Centre Interuniversitaire de Calcul
de Bordeaux (CICBx)

Graphics standards

Software technical Bulletin
SUN, Feb. 88 : 225-251

GKS : la norme graphique

Support de cours
Centre National Universitaire Sud de Calcul
(CNUSC)
José SANCHEZ, 11/05/87

Infographie I et II

Philippe SCHWEIZER
Presses Polytechniques Romandes (ed.), 1987

UNIGKS TIPS

Arne JORGENSEN
UNIRAS (ed.), 1985

GSS - tool kit

Kernel System - level 2 b
Programmer's guide
GRAPHIC SOFTWARE SYSTEM INC. (ed.), 1985

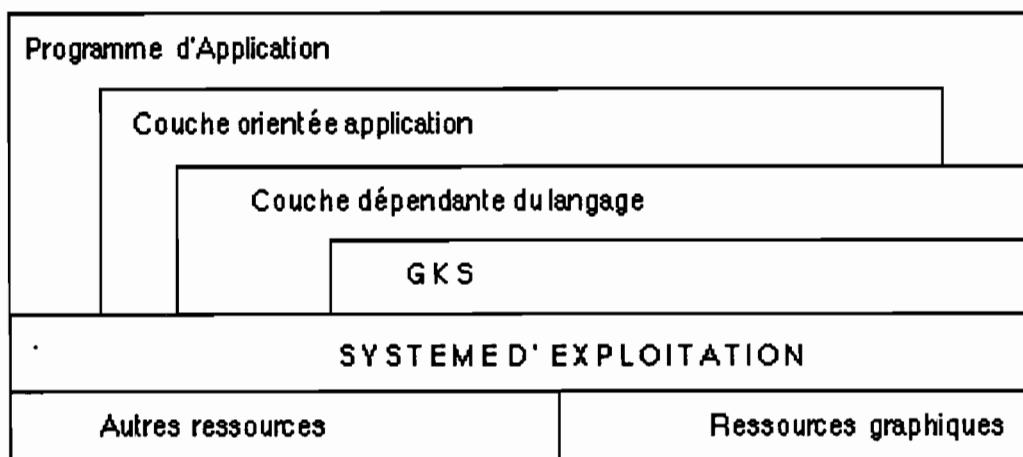
LE NOYAU GRAPHIQUE GKS

1. INTRODUCTION

Le système GKS constitue une interface fonctionnelle entre un programme d'application et une configuration de dispositif d'entrée et de sortie graphique.

L'interface contient toutes les fonctions de base des opérations graphiques interactives et non interactives à d'un grand nombre d'équipements graphiques.

GKS ne définit qu'un noyau graphique indépendant du langage. Pour être intégré à un langage, GKS doit être inclus dans une couche dépendante du langage et contenant ses conventions, par exemple l'affectation des paramètres et des noms.



Modèle en couches GKS

Chaque couche peut appeler les fonctions des couches adjacentes inférieures.

Les principaux concepts de GKS sont :

- les sorties
- les entrées
- les postes de travail
- les transformations
- les segments
- les attributs
- les métatextes

2. LES SORTIES GRAPHIQUES

2.1. LES PRIMITIVES DE SORTIE

Ce sont des abstractions d'actions élémentaires qu'un périphérique peut effectuer. Elles sont au nombre de six.

POLYLIGNE	GKS génère une ligne brisée dont les sommets sont définis par une séquence de points.
POLYMARQUE	GKS génère un symbole centré aux positions données.
TEXTE	GKS génère une chaîne de caractères à la position donnée.
POLYGONE	GKS génère un polygone qui peut être vide ou rempli par une couleur uniforme, un motif ou des hachures.
MATRICE DE CELLULES	GKS génère une matrice de pixels, chacun étant affecté d'une couleur.
PRIMITIVE GRAPHIQUE GENERALISEE	GKS fait appel aux fonctions spéciales de sortie géométrique d'un poste de travail.

2.2. LES ATTRIBUTS DES PRIMITIVES DE SORTIE

Chaque primitive est caractérisée par un ensemble de paramètres ou attributs regroupés en trois classes :

- les attributs géométriques
- les attributs non géométriques
- les identificateurs de désignations (cf. segments)

Une polyligne ne possède pas d'attributs géométriques ; elle est décrite par son type, son épaisseur et sa couleur de trait qui sont liés au poste de travail.

Une polymarque est également dépourvue d'attribut géométrique.

Un texte est par contre géométriquement défini par sa hauteur de lettre, son orientation, son sens d'écriture et son alignement par rapport à une référence. Il est de plus défini par d'autres attributs non géométriques tels que la police, la précision d'écriture, la couleur, etc...

Pour les quatres primitives (POLYLIGNE, POLYMARQUE, TEXTE, POLYGONE), on peut dresser la liste de 13 attributs non géométriques.

- | | |
|----|---------------------------------------|
| 1 | type de trait |
| 2 | coefficient d'épaisseur |
| 3 | index de couleur |
| 4 | type de marque |
| 5 | coefficient de taille |
| 6 | index de couleur |
| 7 | police et précision de texte |
| 8 | facteur d'agrandissement |
| 9 | facteurs d'espacement inter-caractère |
| 10 | index de couleur |
| 11 | style de remplissage du polygone |
| 12 | index de style |
| 13 | index de couleur |

Les attributs non géométriques d'une primitive peuvent être spécifiés de deux façons : par un groupage ou individuellement.

Pour spécifier les aspects par l'intermédiaire des tables de groupage, il existe un attribut par primitive appelé INDEX. Cet attribut est utilisé comme un index d'entrée dans une table de groupage pour chaque primitive sauf pour la Matrice de Cellules et la Primitive Graphique Généralisée.

Chaque poste de travail dispose de son propre jeu de tables de groupage.

Pour spécifier individuellement un paramètre de l'aspect, il existe un attribut individuel propre à chaque aspect non géométrique. De même que les attributs définissant des caractères géométriques, ces attributs sont indépendants du poste de travail et sont mémorisés dans la table d'état de GKS.

Lorsqu'une primitive est affichée, les valeurs des aspects non géométriques utilisés pour l'affichage sont déterminés comme suit :

- a) Si pour un aspect, l'IA est "individuel", la valeur utilisée sur tous les postes de travail est la valeur courante de l'attribut individuel de cette primitive
- b) Si pour un aspect, l'IA est "groupé", la valeur utilisée sur un poste de travail est obtenue par la table de groupage de la primitive et du poste de travail. On utilise la valeur de l'entrée indiquée par l'index de groupage.

2.2.1. Les attributs de POLYLIGNE

INDEX DE POLYLIGNE
TYPE DE TRAIT
COEFFICIENT D'EPATISSEUR DE TRAIT
INDEX DE COULEUR DE LIGNE
IA DE TYPE DE TRAIT
IA DE COEFFICIENT D'EPATISSEUR
IA D'INDEX DE COULEUR
IDENTIFICATEUR DE DESIGNATION

La primitive POLYLIGNE n'a pas d'attribut géométrique. La représentation d'une polyligne sur un poste de travail est contrôlée par la combinaison des attributs groupés et individuels dépendant des valeurs des Indicateurs d'Aspect.

Les types de trait 1 à 4 sont : continu, tireté, pointillé et mixte. D'autres types de traits peuvent être pris en compte mais leurs styles sont alors dépendants de la réalisation.

L'épaisseur du trait est calculée en multipliant l'épaisseur nominale par le coefficient d'épaisseur de trait. Cette valeur est arrondie par le poste de travail à la valeur d'épaisseur disponible la plus proche.

2.2.2. Les attributs de POLYMARQUE

INDEX DE MARQUE
TYPE DE MARQUE
COEFFICIENT DE TAILLE DE MARQUE
INDEX DE COULEUR DE MARQUE
IA DE TYPE DE MARQUE
IA DE COEFFICIENT DE TAILLE
IA D'INDEX DE COULEUR DE MARQUE
IDENTIFICATEUR DE DESIGNATION

La primitive POLYMARQUE n'a pas d'attribut géométrique. La représentation d'une polymarque sur un poste de travail est contrôlée par la combinaisons des attributs groupés et individuels dépendant des valeurs de Indicateurs d'Aspect.

Les types de marque 1 à 5 sont : le point, le signe plus, l'astérisque, le cercle et la croix en diagonale centrés chacun au point qu'il désigne. D'autres types de marques peuvent être pris en compte mais leurs formes dépendent de la réalisation.

La taille de la marque est calculée en multipliant la taille nominale par le coefficient de taille de marque. La taille est arrondie pour le poste de travail à la taille disponible la plus proche. La marque de type 1 est toujours affichée comme le plus petit point affichable.

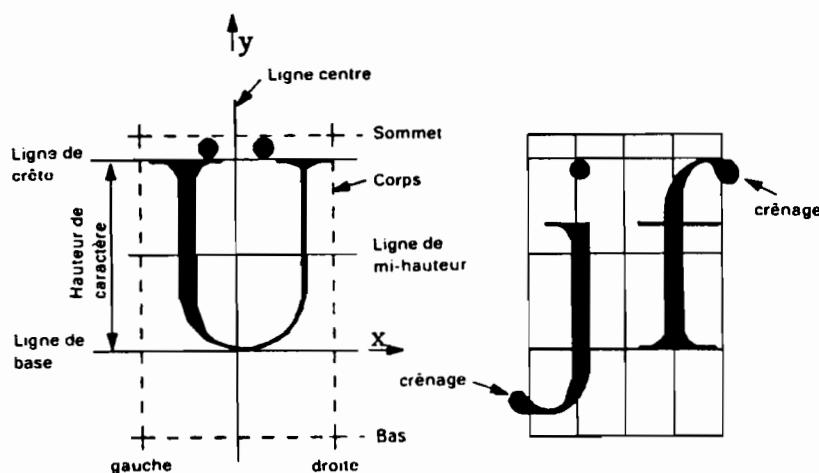
2.2.3. Les attributs du TEXTE

INDEX DE TEXTE
 POLICE ET PRECISION DU TEXTE
 FACTEUR D'AGRANDISSEMENT DU TEXTE
 ESPACEMENT INTER-CARACTERE
 INDEX DE COULEUR DE TEXTE
 IA DE POLICE ET PRECISION
 IA DE FACTEUR D'AGRANDISSEMENT
 IA D'ESPACE INTER-CARACTERE
 IA D'INDEX DE COULEUR DU TEXTE
G HAUTEUR DE CARACTERE
G VECTEUR HAUTEUR DES CARACTERES
G SENS D'ECRITURE
G ALIGNEMENT DU TEXTE
 IDENTIFICATEUR DE DESIGNATION

Le texte possède aussi deux attributs géométriques spécifiés de manière implicite : la LARGEUR DU CARACTERE et le VECTEUR BASE DE CARACTERE. Ils se comportent exactement comme les autres attributs géométriques. Les deux attributs sont modifiés implicitement par les fonctions SET CHARACTER HEIGHT et SET CHARACTER UP VECTOR.

Le contrôle précis de l'aspect du texte sur un poste de travail est assuré par les éléments suivant : HAUTEUR, LARGEUR, FACTEUR D'AGRANDISSEMENT, SENS D'ECRITURE, VECTEUR HAUTEUR, VECTEUR BASE, ESPACEMENT ET ALIGNEMENT. L'emploi de ces valeurs dépend du choix de l'élément (police, précision).

La hauteur est la hauteur nominale d'une lettre majuscule. La largeur est la largeur nominale attachée à une police, mais la largeur effective de chaque caractère peut varier en fonction des choix du concepteur de la police. Le facteur d'agrandissement est un coefficient d'affinité appliqué au texte à afficher. Il modifie le rapport hauteur/largeur choisi par le dessinateur de la fonte. Le vecteur hauteur précise la direction de l'axe vertical, le vecteur base fournit celle de la ligne de base du caractère. La longueur de ces vecteurs n'est pas significative ; les deux vecteurs transforment le rectangle enveloppe du caractère en un parallélogramme.



Le sens d'écriture droite (0), gauche (1), haut (2) et bas (3) définit la façon dont les cellules de caractère sont juxtaposées. L'espacement, exprimé par une fraction de la hauteur nominale de la police donne l'intervalle qui sera intercalé entre deux cellules consécutives dans le sens de l'écriture. Cet intervalle peut être négatif. Enfin l'alignement horizontal détermine la position relative du texte par rapport à la position de référence de la primitive. Il peut prendre 4 valeurs : normal (0), gauche (1), milieu (2) et droite (3). De même pour l'alignement vertical, position du texte sur l'axe vertical, six valeurs sont possibles : normal (0), sommet (1), crête (2), mi-hauteur (3), base (4) et bas (5).

Chaque composante de l'attribut alignement peut prendre la valeur NORMAL. Si l'alignement est déclaré (NORMAL,NORMAL), il est interprété différemment suivant le sens d'écriture :

SENS	Alignement Horizontal et Vertical ou NORMAL
DROIT	(GAUCHE, BASE)
GAUCHE	(DROIT, BASE)
HAUT	(MILIEU, BASE)
BAS	(MILIEU, MILIEU)

Les valeurs initiales des attributs géométriques sont :

HAUTEUR	WC	0.01 (soit 1 % de la hauteur de la fenêtre par défaut)
VECTEUR HAUTEUR	WC	(0,1)
SENS		DROIT
ALIGNEMENT		(NORMAL,NORMAL)

Les valeurs initiales, les attributs géométriques spécifiés implicitement sont :

LARGEUR	WC	0.01 (soit la même valeur que la valeur initiale de HAUTEUR)
VECTEUR BASE	WC	(0,1)

Police et précision constituent ensemble un attribut. Chaque poste de travail doit disposer d'au moins une police (n°1) Les autres polices sont dépendantes de la réalisation. La valeur de la précision permet d'associer un niveau de fidélité à la représentation du texte sur le poste de travail.

Trois valeurs sont possibles (dans l'ordre croissant du nombre d'attributs pris en compte) :

- CHAINE (string)
- CARACTERE (char)
- HAUTE FIDELITE (stroke)

Puisque les valeurs de HAUTEUR, LARGEUR et des VECTEURS HAUT et BASE sont exprimées en unités de coordonnées universelle (WC), mais que les caractères sont générés sur le poste de travail en coordonnées d'appareil en police propre au poste, les attributs géométriques doivent subir des transformations afin que le poste génère les caractères de la façon prévue

2.2.4. Les attributs du POLYGONE

	INDEX DE POLYGONE
	STYLE DE REMPLISSAGE DU POLYGONE
	INDEX DE STYLE DE POLYGONE
	INDEX DE COULEUR DU POLYGONE
	IA DE STYLE DE REMPLISSAGE DE POLYGONE
	IA D'INDEX DE STYLE DU POLYGONE
	IA d'INDEX DE COULEUR DU POLYGONE
	TAILLE DU MOTIF
G	POINT DE REFERENCE DU MOTIF
	IDENTIFICATEUR DE DESIGNATION

Le seul attribut géométrique est le point de référence du motif. Il possède cependant 2 attributs géométriques implicites HAUTEUR DU MOTIF et LARGEUR DU MOTIF qui sont définis par la fonction SET PATTERN SIZE.

La représentation d'un polygone sur un poste de travail est contrôlée par la combinaison des attributs groupés et individuels dépendant des valeurs des Indicateurs d'Aspect.

Le style de remplissage peut prendre les valeurs suivantes :

- | | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> - CONTOUR (hollow) - PLEIN (solid) - MOTIF (pattern) - HACHURE (hatch) | <ul style="list-style-type: none"> Pas de remplissage
seul le contour apparaît L'intérieur du polygone
est colorié uniformément L'intérieur du polygone
est rempli par un motif L'intérieur du polygone
est rempli par des hachures |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

3. LES POSTES DE TRAVAIL

Le poste de travail (workstation) est un concept majeur dans le système GKS. C'est un interface logique qui permet aux programmes d'application de contrôler les appareils physiques. Dans toute réalisation de GKS pour chaque type de poste de travail, il existe une table de description qui spécifie les caractéristiques et les possibilités du poste. Un programme d'application pourra, par des fonctions d'interrogation, adapter son comportement aux possibilités du ou des postes de travail qu'il exploite.

Un poste de travail graphique idéal :

- possède une surface d'affichage adressable
- autorise seulement les espaces d'affichage rectangulaire
- permet l'emploi de sous-espaces d'affichages
- supporte plusieurs polices de caractères, plusieurs types de traits
- a au moins un appareil logique d'entrée pour chaque classe d'entrée
- permet les entrées de type REQUETE, ECHANTILLONNAGE et EVENEMENT
- stocke les segments et permet de les manipuler

Dans la pratique, toutes ces qualités ne sont que partiellement exploitables

On distingue six catégories de poste de travail :

SORTIE	(output)
ENTREE	(input)
INTERACTIF	(outin)
WISS	Poste de stockage de segments indépendants des postes de travail
MO	Métafichier en sortie
MI	Métafichier en entrée

GKS permet à l'aspect des primitives graphiques de sortie de varier d'un poste de travail à l'autre, afin de tirer éventuellement profit de leurs possibilités propres. Des résultats différents pourront être obtenus pour les représentations des polylinéaires, polymarques, textes, polygones, motifs et couleurs, la gestion d'état différé et la transformation de poste de travail.

Comment sélectionner un poste de travail ?

Dans son programme d'application, l'utilisateur choisit un poste de travail, par l'intermédiaire d'un identificateur. L'ouverture du poste de travail permet d'associer l'identificateur du poste de travail à un numéro de connexion (sous Fortran, ce sera une unité logique). L'état courant du poste de travail est conservé dans la liste d'état des postes de travail.

Il est alors possible sur ce poste ouvert de manipuler des segments, de réaliser des opérations d'entrées, et de définir des attributs. Pour que les primitives graphiques soient transmises, il faut que le poste de travail soit actif.

Un poste de travail actif peut être désactivé puis fermé.

Effacement de la surface de travail

Dans la liste d'état du poste de travail, il existe une entrée "surface d'affichage vide". Cette entrée est positionnée à l'état NON VIDE dès qu'une primitive de sortie a été transmise.

Deux possibilités d'effacement sont offertes :

- a) effacer la surface impérativement
- b) effacer la surface que si c'est nécessaire

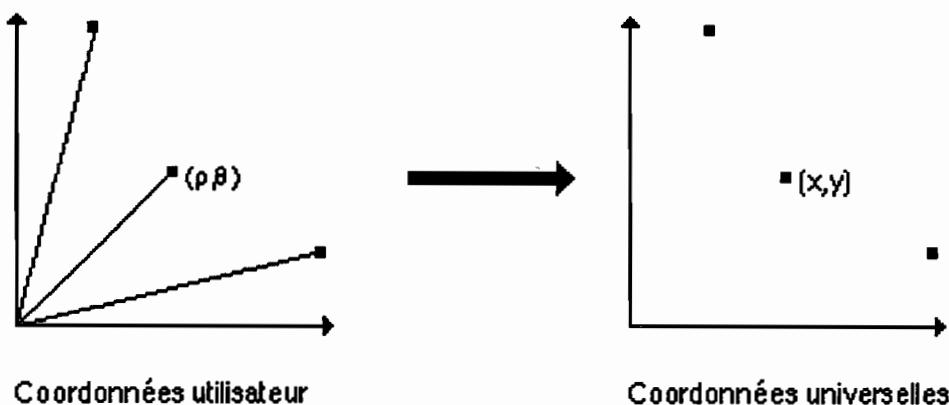
4. LES TRANSFORMATIONS

GKS utilise trois systèmes de coordonnées :

- coordonnées universelles
- coordonnées normées
- coordonnées d'appareil

4.1. COORDONNEES UTILISATEURS ET COORDONNEES UNIVERSELLES

L'utilisateur utilise de préférence un système de coordonnées adapté à son application : système logarithmique, système scalaire. Les seuls systèmes acceptés par GKS doivent être cartésiens. Il sera de la responsabilité de l'utilisateur de se ramener à un tel système.



Le système (wc : World Coordinates) est exprimé dans les unités du problème et les graduations sont linéaires. Ce sera ce système qui sera utilisé pour décrire les primitives de GKS.

4.2. COORDONNEES NORMEES D'APPAREIL

Le système de coordonnées normées d'appareil (NDC : Normalized Device Coordinates) est un repère orthonormé limité par le carré $(0,1) \times (0,1)$. Il représente une unité virtuelle d'affichage où une image est éventuellement écrêtée ou découpée ("clipping"), manipulée ("segment") ou archivée ("métafichier").

4.3. COORDONNEES D'APPAREIL

Le système de coordonnées d'appareil (DC : Device Coordinates) est un repère orthogonal - obligatoirement normé - exprimé en unités de dispositif d'affichage (mètres, centimètres, pixels).

GKS définit les lois de transformation qui permettent de projeter l'espace utilisateur sur le périphérique virtuel (transformation de normalisation) et l'espace normalisé sur l'espace d'affichage réel (transformation du poste de travail).

4.4. TRANSFORMATION DE NORMALISATION

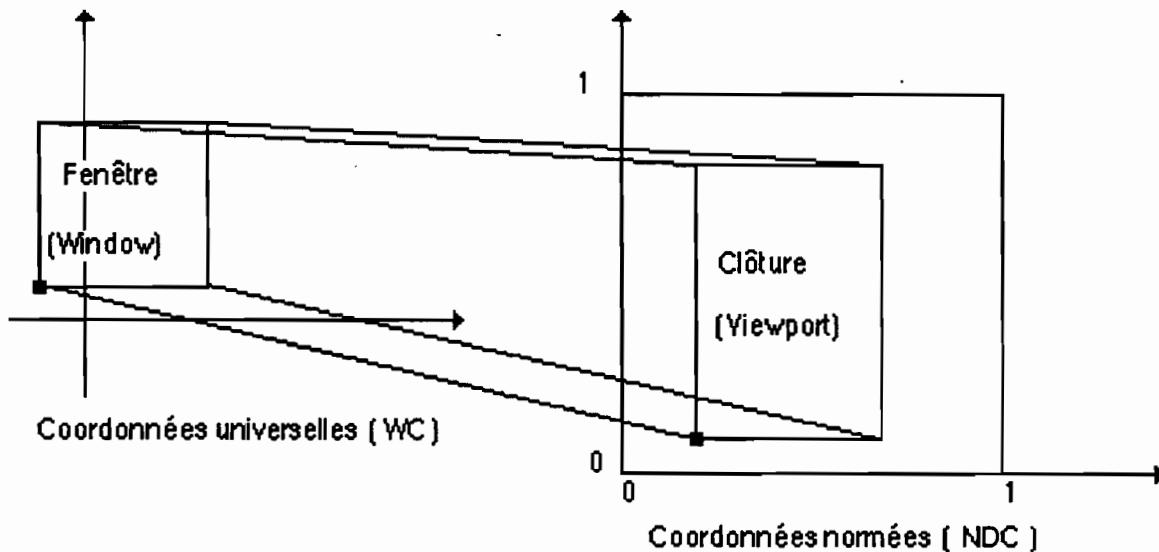
Cette transformation est définie en spécifiant :

- les limites d'une zone rectangulaire (bord parallèles aux axes) dans l'espace WC. C'est la fenêtre de la transformation de normalisation (WINDOW)
- les limites d'une zone rectangulaire dans l'espace NDC. C'est la clôture de la transformation de normalisation (VIEWPORT)

Les frontières des deux rectangles sont incluses

Cette transformation comprend notamment une mise à l'échelle avec des facteurs d'échelle positifs qui peuvent être différents pour les deux axes. La transformation est identifiée par un numéro compris entre 0 et n. La valeur maximale de n dépend de la réalisation. La transformation 0 est réservée à la transformation unitaire.

On peut définir plusieurs transformations de normalisation au sein d'une même application. On sélectionne l'une d'entre elles par son numéro. Par défaut, c'est la transformation 0 qui est sélectionnée.



Découpage

La sélection d'une transformation de normalisation définit un rectangle de découpage de l'image. L'entrée "rectangle de découpage" dans la table d'état de GKS est remplie avec la clôture de la transformation. Le mécanisme de découpage peut être rendu actif ou non (indicateur de découpage).

Le découpage est retardé jusqu'à l'affichage des primitives graphiques sur la surface de visualisation du poste de travail. Par conséquent une primitive graphique envoyée dans un segment est transformée en unités NDC, et un rectangle de découpage y est associé.

L'insertion de segment permet de modifier le rectangle de découpage stocké avec les primitives graphiques lors de la création du segment.

Les primitives envoyées à un poste de type MO (Metafile Output) ne sont pas découpées.

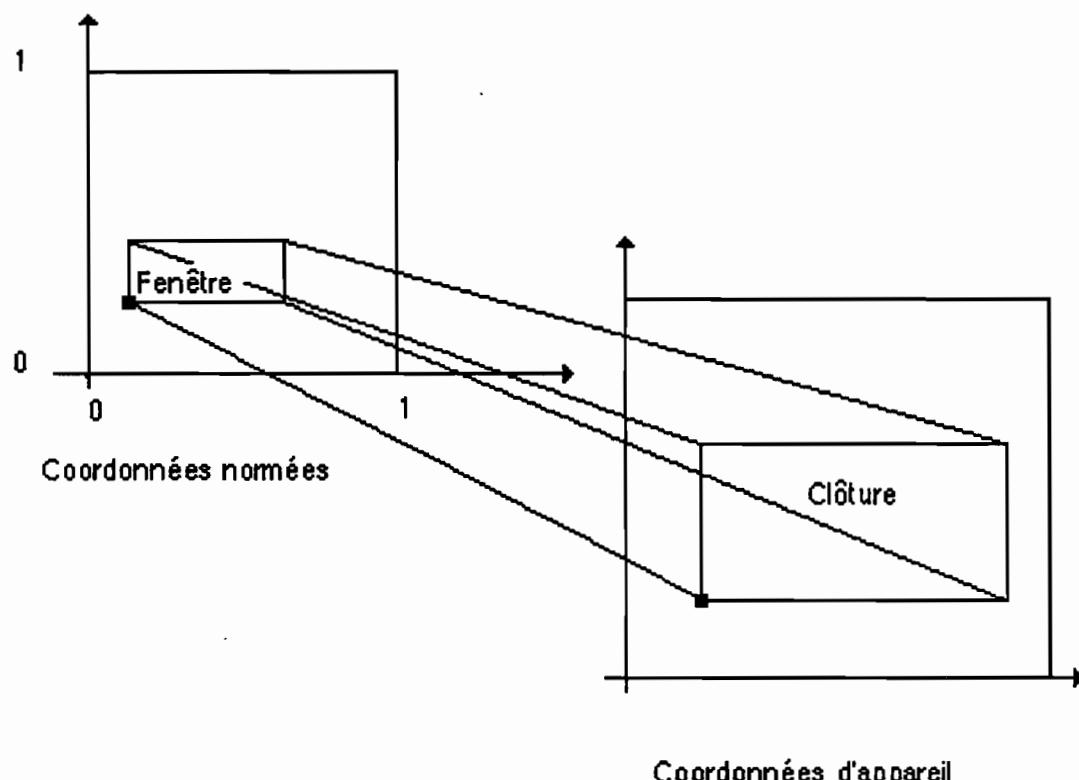
4.5. TRANSFORMATION DE POSTE DE TRAVAIL

L'espace NDC (surface abstraite d'affichage) est unique. Les postes de travail ouverts peuvent choisir une partie de cet espace pour l'afficher sur leur dispositif de visualisation. A un poste de travail peut être associée une transformation de l'espace NDC vers l'espace DC de celui-ci.

Le système de coordonnées d'appareil se projette sur la surface d'affichage de la manière suivante :

- a) l'origine des coordonnées d'appareil est le coin bas gauche de l'espace d'affichage
- b) les unités de l'espace de coordonnées d'appareil sont reliées aux coordonnées d'affichage de telle façon qu'un carré dans l'espace de coordonnées d'appareil apparaît comme un carré sur la surface d'affichage
- c) x et y croissent respectivement vers la droite et vers le haut

Les limites de l'espace d'affichage sont stockées dans la table de description du poste de travail.



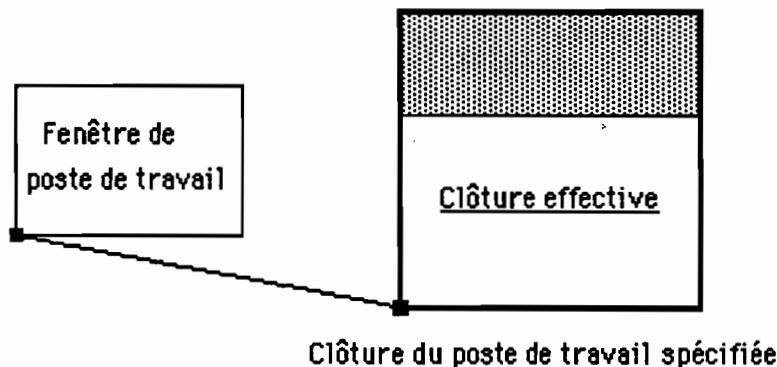
Cette transformation met en oeuvre notamment une mise à l'échelle. Le **facteur d'échelle** (positif) est cette fois **unique**.

Cette transformation est définie en spécifiant :

- les limites d'une zone rectangulaire dans l'espace NDC, c'est la fenêtre de la transformation de poste de travail
- les limites d'une zone rectangulaire dans l'espace DC, c'est la clôture de la transformation de poste de travail

Les frontières sont incluses. Afin qu'aucune sortie se fasse en dehors de la fenêtre du poste de travail, GKS découpe impérativement suivant les limites de celle-ci.

La spécification des deux rectangles peut conduire à définir des ratios différents sur les deux axes le facteur d'échelle étant unique, la transformation de poste de travail projette la fenêtre dans le plus grand rectangle inclus dans la clôture. Le coin bas gauche de la fenêtre est toujours projeté dans le coin bas gauche de la clôture. Par conséquent, un espace est éventuellement laissé inutilisé en haut ou à droite dans la clôture du poste de travail.



5. LES SEGMENTS

Un segment est un ensemble de primitives graphiques manipulables comme un tout. Chaque segment est identifié par un nom unique.

Les segments peuvent être :

- transformés
- rendus visibles ou invisibles
- mis en évidence
- affectés d'un niveau de priorité
- rendus détectables ou indétectables
- supprimés
- renommés
- insérés dans le segment ouvert

A chaque primitive contenue dans un segment est associé un identificateur de désignation (PICK IDENTIFIER).

5.1. ARCHIVAGE DES SEGMENTS :

- dans la zone de stockage de segment dépendant du poste de travail : WDSS (WDSS : Workstation Device Segment Storage)
- dans un poste de travail particulier : WISS (WISS : Workstation Independant Segment Storage)
Ce poste est unique pour toute réalisation de GKS

Les segments stockés dans le WISS peuvent être copiés sur un poste de travail, associés à un poste de travail ou insérés éventuellement dans un autre segment.

A l'inverse, il n'est pas possible de transmettre un segment stocké dans un WDSS vers un autre poste de travail.

5.2. GESTION DES SEGMENTS

Un segment, dès l'instant de sa création, reçoit l'ensemble des primitives générées par l'application à destination des postes de travail ouvert. On peut y insérer des segments déjà existants dans le WISS. Après sa fermeture, il devient impossible d'y ajouter ou d'y soustraire quoi que ce soit. Dans l'état fermé un segment peut être détruit.

5.3. ATTRIBUTS DES SEGMENTS

Transformation : A chaque segment est attaché une matrice (2,3) de transformation (mise à l'échelle, rotation, translation).

Visibilité : un segment est affiché ou non.

Mise en évidence : un segment visible est mis en évidence ou non.

Priorité : lors de l'affichage les segments de priorité la plus haute seront préférés.

Détectabilité : le segment pourra être sélectionné ou non par une primitive d'entrée.

6. LES ENTREES GRAPHIQUES

L'opérateur doit pouvoir agir sur le comportement de son application GKS fournit au moins un appareil d'entrée, lequel appartient obligatoirement à un poste de travail d'entrée ou interactif.

Un appareil d'entrée est identifié par :

- un identificateur de poste de travail
- une classe d'entrée
- un numéro d'appareil

De même que pour les sorties graphiques pour lesquelles on n'a pas à se préoccuper des caractéristiques physiques d'affichage, les appareils d'entrée sont répartis en six classes de dispositifs logiques.

1) RELEVEUR DE COORDONNEES (locator) :

fournit une position en coordonnées universelles et un numéro de transformation de normalisation

2) RELEVEUR D'UNE SUITE DE COORDONNEES (stoke) :

fournit une séquence de points (WC) et un numéro de transformation de normalisation

3) ENTREE DE SCALAIRES (valuator) :

fournit un nombre réel

4) SELECTEUR (choice) :

fournit un état de choix et un entier non négatif qui représente une sélection parmi un certain nombre de choix

5) DESIGNATION (pick) :

fournit un état de désignation, un nom de segment et un identificateur de désignation Une primitive située en dehors d'un segment ne peut être désignée

6) ENTREE DE CHAINE DE CARACTERES (string) : fournit une chaîne de caractères

Le contexte de fonctionnement d'un appareil logique d'entrée est défini par un certain nombre de paramètres et de dispositifs qui lui sont associés. On caractérise un appareil d'entrée par un mesure, une gâchette (trigger), une valeur initiale, un type d'écho/notification, une zone d'écho et un enregistrement de données contenant les caractéristiques du type d'écho et de notification.

La mesure et la gâchette sont des caractéristiques du poste de travail. Les autres éléments caractéristiques peuvent être fournis par l'application.

GKS définit trois modes d'interaction :

- mode requête (niveau b) (un seul processus est actif)
- mode échantillonnage (traitement et acquisition sont simultanés)
- mode événement (simultanéité et file d'attente des mesures)

7. LES METAFICHIERS

GKS propose la possibilité de stocker les informations graphiques ou non dans des fichiers séquentiels appelés métafichiers.

L'utilisation de métafichiers permet notamment :

- le transport des informations graphiques d'un ordinateur à un autre (réseau, support magnétiques, etc)
- le transfert des informations graphiques d'une application GKS à une autre,
- l'archivage en vue d'une utilisation ultérieure

Les métafichiers GKS se comportent comme des postes de travail. On distingue deux types de métafichiers :

- les métafichiers en entrée (MI, metafile input)
- les métafichiers en sortie (MO, metafile output)

A partir de l'instant où un poste de travail de type métafichier en sortie est ouvert, puis activé, toutes les sorties de GKS y sont mémorisées jusqu'à la désactivation de ce poste. Si des informations non graphiques doivent être inscrites dans ce métafichier, la fonction "WRITE ITEM TO GKSM" permet de réaliser cette insertion.

Pour exploiter un métafichier existant, il faut ouvrir un poste de travail de type métafichier en entrée. Ce poste ne doit pas être activé. Le premier enregistrement du fichier devient "l'enregistrement courant". La procédure d'exploitation des enregistrement est la suivante :

- obtention des caractéristiques de l'enregistrement courant (GET ITEM FROM GKSM)
- transfert de l'enregistrement de données dans une zone mémoire de l'application (READ ITEM FROM GKSM), l'enregistrement suivant devient le nouvel enregistrement suivant
- interprétation de l'item acquis (INTERPRET ITEM)

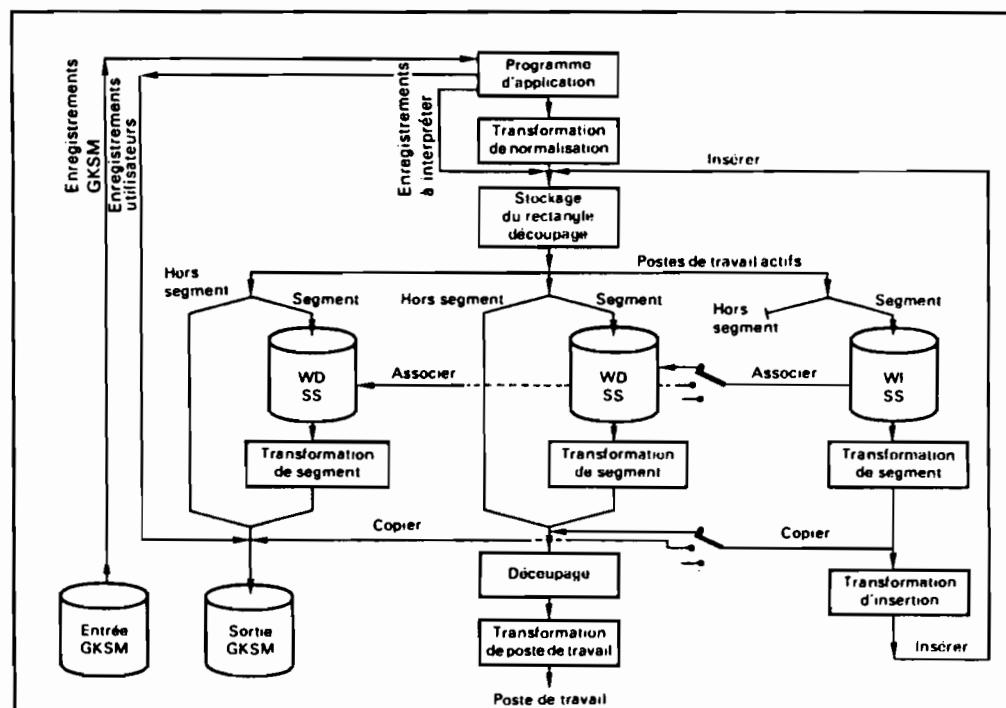
Remarques

Pour qu'un dessin stocké dans un métacharier puisse être réaffiché ultérieurement sur une unité physiquement différente, et pour laquelle la transformation de poste de travail initiale n'a pas forcément de sens, les images archivées sont exprimées en coordonnées (NDC). Aucun découpage n'est effectué, évitant ainsi toute perte d'information. Cependant les rectangles de découpage (NDC) associés aux différents objets graphiques sont également enregistrés.

Chaque image est représentée par une série d'enregistrements de données ('item'). Pour l'instant, il n'existe pas de norme de création des métachiers. Cependant un fichier est constitué généralement d'un en-tête, d'une suite d'enregistrements et d'un bloc de fin. Chaque enregistrement est composé d'un en-tête optionnel ('GKSM'), d'un identificateur des données qu'il contient (les numéros d'identification de 1 à 100 sont normés), d'une longueur exprimée en caractères et des données proprement dites. Ces enregistrements sont de longueur variable.

8. CONCLUSION

L'ensemble complet des flots de données pour les sorties graphiques est représenté par le diagramme ci-dessous.

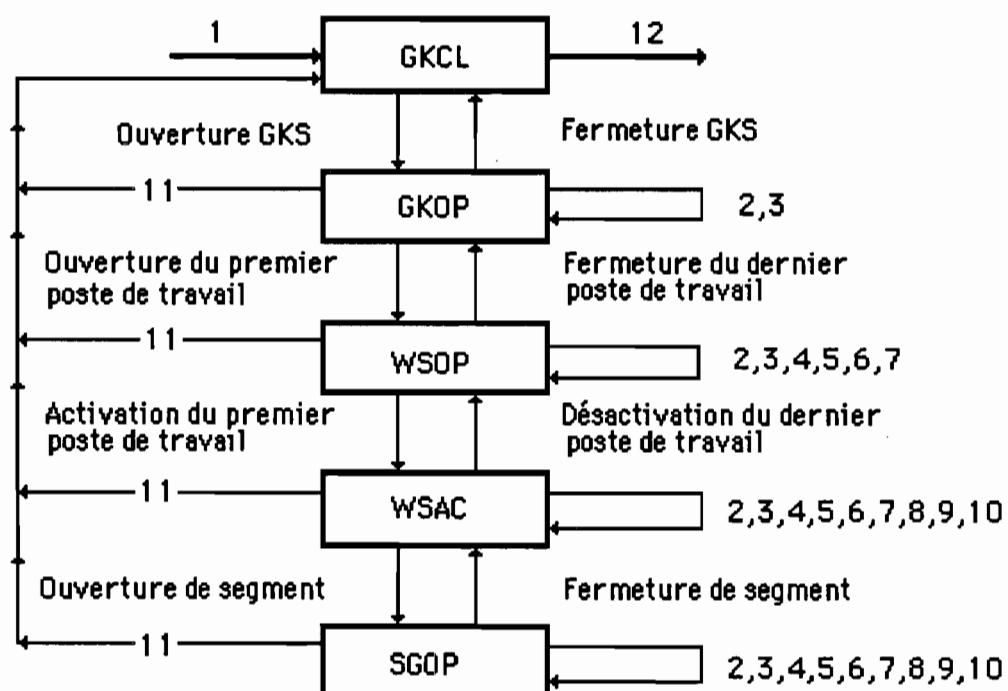


GKS peut se trouver dans l'un des cinq états suivants :

GKCL	GKS fermé
GKOP	GKS ouvert
WSOP	Un poste de travail ouvert au moins
WSAC	Un poste de travail actif au moins
SGOP	Segment ouvert

L'état de GKS est repéré par la valeur d'une variable globale, initialisée avant le premier appel à la valeur GKCL.

Automate de transition entre les états de GKS



- | | |
|-----------------------------------|---------------------------------------|
| 1 Début du programme | 7 Entrées graphiques |
| 2 Interrogation des tables de GKS | 8 Activation d'un poste de travail |
| 3 Définition d'attribut | 9 Désactivation d'un poste de travail |
| 4 Ouverture d'un poste de travail | 10 Primitives de sorties |
| 5 Fermeture d'un poste de travail | 11 Fermeture d'urgence |
| 6 Manipulation de segments | 12 Fin du programme |

L'ensemble complet des variables d'état de GKS est composé des sous-ensembles suivants :

- 1) état de GKS,
- 2) liste d'état de GKS,
- 3) liste d'état de chacun des segments,
- 4) file d'attente des entrées,
- 5) liste d'état de chacun des postes de travail ouverts
- 6) liste d'état des erreurs de GKS.



DESCRIPTION DE CERTAINES PROCEDURES GKS

⇒ **OPEN GKS**

Ouverture de GKS



CALL GOPKS(NUMFIC,IDUM)

PARAMETRES :

NUMFIC En, I Unité logique du fichier des erreurs
IDUM En, I

EFFET : - Le système GKS est mis à l'état GKOP
- La liste d'état de GKS est allouée
- Les tables de GKS sont accessibles

EXEMPLE :

```
character * 20 ficerr  
  
data luerr,ficerr / 16,'gkserr' /  
  
open (luerr,file = ficerr,...)  
rewind luerr  
idum = 1  
call gopks (luerr,idum)
```

⇒ **CLOSE GKS**

Fermeture de GKS



CALL GCLKS

EFFET : - Le système GKS est mis à l'état GKCL
- La liste d'état de GKS et les tables de description
- Tous les fichiers de GKS sont fermés

NOTE : - GKS peut être ouvert à nouveau

⇨ **OPEN WORKSTATION** Ouverture d'un poste de travail 

CALL GOPWK(IDPT, IDCON, ITYPT)

PARAMETRES :

IDPT	En I	Identificateur du poste de travail
IDCON	En I	Identificateur de connexion (ul. fichier)
ITYPT	En I	Type de poste de travail (driver)

EFFET :

- Le système GKS est mis à l'état WSOP
- La liste d'état du poste de travail est allouée et initialisée
- L'identificateur du poste de travail est ajouté à l'ensemble des postes ouverts dans la liste d'état de GKS
- GKS demande au système d'exploitation d'établir la connexion spécifiée

NOTE :

- La surface d'affichage (console) est effacée

EXEMPLE :

```
character * 20 ficwk  
  
data luwk, ficwk / 20, 'bidon' /  
  
idpt = 1  
open (luwk,file=ficwk,...)  
rewind luwk  
  
write (*,10)  
10 format (' Poste de travail ? ', $)  
read (*,*) itypt  
  
call gopwk (idpt,luwk,itypt)
```

GKS _{Bx} (itypt) : 6971	Ecran EGA (PC, MS/DOS)
: 7280	Plotter HPGL A4
: 7374	Ecran Sun monochrome
: 7375	Ecran Sun couleur
:	
: 1000	Imprimante laser Postscript

⇨ **CLOSE WORKSTATION** Fermeture d'un poste de travail 

CALL GCLWK(IDPT)

EFFET :

- La liste d'état du poste spécifié est libérée après que toutes les primitives de sortie soient terminées
- Si l'ensemble des postes associés à un segment devient vide, ce segment est supprimé
- La connexion avec le poste de travail est libérée.
- GKS est mis dans l'état GKOP si aucun poste de travail ne demeure ouvert

NOTE :

- La surface d'affichage ne doit pas être nécessairement effacée, mais elle peut l'être

⇨ **ACTIVATE WORKSTATION** Activation d'un poste de travail

CALL GACWK(IDPT)

EFFET :

- Le poste de travail est marqué actif
- L'identificateur du poste est ajouté à la liste des postes actifs c.a.d. ceux qui reçoivent toutes les primitives de sortie
- Si GKS est dans l'état WSOP, il est mis à l'état WSAC

⇨ **DESACTIVATE WORKSTATION** Désactivation d'un poste de travail

CALL GDAWK(IDPT)

EFFET :

- L'entrée état de la liste d'état du poste de travail est marquée inactive
- L'identificateur du poste est retiré de la liste des postes actifs dans la liste d'état de GKS

NOTE :

- Les primitives ne sont plus transmises à ce poste, il ne stocke plus de nouveaux segments, par contre les segments déjà stockés sont conservés

⇨ **CLEAR WORKSTATION** Nettoyage du poste de travail

CALL GCLRWK(IDPT,ICONT)

PARAMETRES :

IDPT En, I Identificateur du poste de travail
ICONT En, I Identificateur de contrôle

EFFET : - Toutes les actions différences pour le poste spécifié sont exécutées.
 - La surface d'affichage est mis à l'état libre et est,
 soit toujours effacée (ICONT = 1), soit conditionnellement
 effacée (ICONT = 0). La condition est : l'entrée "surface
 d'affichage vide" dans la liste d'état du poste est à l'état NON VIDE.

⇨ **MESSAGE** Envoi d'un message 

CALL GMSG(IDPT,MSG)

PARAMETRES :

IDPT En, I Identificateur du poste de travail
MSG En, C Message

EFFET : - Affiche un message à un endroit dépendant
 de la mise en oeuvre ou sur un autre appareil associé au poste de
 travail

⇨ SET WINDOW

Définition de la fenêtre d'une transformation de normalisation

CALL GSWN(NUMTR,XWMIN,XWMAX,YWMIN,YWMAX)

PARAMETRES :

NUMTR	En, I	numéro de la transformation de normalisation
XWMIN	En, R	abscisse point inférieur gauche (WC)
XWMAX	En, R	abscisse point supérieur droit (WC)
YWMIN	En, R	ordonnée point inférieur gauche (WC)
YWMAX	En, R	ordonnée point supérieur droit (WC)

EFFET :

- Les limites de la fenêtre de la transformation spécifiée sont mises aux valeurs spécifiées dans l'entrée correspondante de la liste d'état de GKS

NOTE :

- Par défaut, la transformation 0 est associée à la fenêtre $(0,1) \times (0,1)$

⇨ SET VIEWPORT

Définition de la clôture d'une transformation de normalisation

CALL GSVP(NUMTR,XVMIN,XVMAX,YVMIN,YVMAX)

PARAMETRES :

NUMTR	En, I	numéro de la transformation de normalisation
XVMIN	En, R	abscisse point inférieur gauche (NDC)
XVMAX	En, R	abscisse point supérieur droit (NDC)
YVMIN	En, R	ordonnée point inférieur gauche (NDC)
YVMAX	En, R	ordonnée point supérieur droit (NDC)

EFFET :

- Les entrées correspondantes de la liste d'état de GKS sont mises aux valeurs spécifiées. Le rectangle de découpage dans la liste d'état de GKS est remplacé par les limites de la clôture si le numéro de transformation est le même que celui de la transformation courante.

NOTE :

- Par défaut la clôture associée à la transformation 0 est $(0,1) \times (0,1)$

⇒ **SET VIEWPORT INPUT PRIORITY** Définition de la priorité relative d'une clôture

CALL GSVPIP(NTRN,NTREF,IPRI)

PARAMETRES :

NTRN En, I numéro de la transformation de normalisation
NTREF En, I numéro de la transformation de référence
IPRI En, I indicateur (priorité relative)
 0 : la clôture de la transformation NTRN à une priorité supérieure
 1 : la clôture de la transformation NTRN à une priorité inférieure

EFFET : - La priorité d'entrée de la clôture de la transformation de normalisation spécifiée devient supérieure ou inférieure à celle de la transformation de référence. La liste d'état de GKS est mise à jour.

NOTE : - Les autres transformations conservent leur priorité

⇒ **SELECT NORMALIZATION TRANSFORMATION**

Définition de la transformation de normalisation courante

CALL GSELNT(NUMTR)

EFFET : - L'entrée "numéro de transformation de normalisation courante" dans la liste d'état de GKS est mise à la valeur spécifiée. Le rectangle de découpage est remplacé par les limites de la clôture de la transformation indiquée

⇒ **SET CLIPPING INDICATOR** Activation/désactivation de l'écrêtage 

CALL GSCLIP(ICLIP)

PARAMETRES :

ICLIP En, I indicateur d'écrêtage
 0 : pas de découpage
 1 : découpage (valeur par défaut)

EFFET : - L'entrée "indicateur de découpage" dans la liste d'état de GKS est mise à la valeur spécifiée

⇨ SET WORKSTATION WINDOW

Définition de la fenêtre de la transformation du poste de travail

CALL GSWKWN(IDPT,XMIN,XMAX,YMIN,YMAX)

PARAMETRES :

IDPT	En, I	identificateur du poste de travail
XMIN	En, R	abscisse du point inférieur gauche (NDC)
XMAX	En, R	abscisse du point supérieur droit (NDC)
YMIN	En, R	ordonnée du point inférieur gauche (NDC)
YMAX	En, R	ordonnée du point supérieur droit (NDC)

EFFET : - L'entrée "fenêtre du poste de travail demandée" de la liste d'état du poste de travail est mise à la valeur spécifiée

NOTE : - La fenêtre du poste de travail doit être définie dans l'espace (0,1) x (0,1). Par défaut la fenêtre du poste de travail est le carré (0,1) x (0,1)

⇨ SET WORKSTATION VIEWPORT

Définition de la clôture de la transformation du poste de travail

CALL GSWKVP(IDPT,XVMIN,XVMAX,YVMIN,YVMAX)

PARAMETRES :

IDPT	En, I	identificateur du poste de travail
XVMIN	En, R	abscisse du point inférieur gauche (DC)
XVMAX	En, R	abscisse du point supérieur droit (DC)
YVMIN	En, R	ordonnée du point inférieur gauche (DC)
YVMAX	En, R	ordonnée du point supérieur droit (DC)

EFFET : L'entrée "clôture de poste de travail demandée" dans la liste d'état du poste de travail est mise à la valeur spécifiée

NOTE : Par défaut la clôture du poste de travail est définie par le carré (0,1) x (0,1)

↔ POLYLINE

Trace de ligne



CALL GPL(NP,PX,PY)

PARAMETRES :

NP	En, I	nombre de points
PX	En, R	abscisses des points
PY	En, R	ordonnées des points

EFFET :

- Une suite de segments de droite est générée en commençant au premier point et en finissant au dernier point. Les valeurs courantes des attributs de polygone sont attachées à la primitive.

ATTRIBUTS :

(*) ASF(.)

Index de polygone courante		CALL GSPLI(INDEX)	En, I
Type de trait	1	CALL GSLN(ITYPE)	En, I
Coefficient d'épaisseur de trait	2	CALL GSLWSC(EPAIS)	En, R
Index de couleur	3	CALL GSPLCI(ICOUL)	En, I

CALL GSPLR(IDPT,INDEX,ITYPE,EPAIS,ICOUL)

Le type de trait est défini comme suit :

< 0	dépend de la réalisation
1	trait plein
2	tireté
3	pointillé
4	mixte
≥ 5	réservé à l'enregistrement ou dépend du matériel

(*) Aspect Source Flag ASF (i)=0 référence à la table de groupage du poste
 ASF (i)=1 référence individuelle

⇒ **POLYMARKER**

Tracé de marques



CALL GPM(NP,PX,PY)

PARAMETRES :

Les mêmes que pour la polyligne

EFFET : - Visualise la position des points en générant une suite de marques.
Les valeurs courantes des attributs de polymarque sont attachées à la primitive.

ATTRIBUTS :

(*) ASF(.)

Index de polymarque courant		CALL GSPMI(INDEX)	En, I
Type de marque	4	CALL GSMK(ITYPE)	En, I
Coefficient de taille de marque	5	CALL GSMSC(DIM)	En, R
Index de couleur	6	CALL GSPMCI(ICOUL)	En, I

CALL GSPMR(IDPT,INDEX,ITYPE,DIM,ICOUL)

Le type de marque est défini par :

< 0	dépend de la réalisation
1	. point
2	plus
3	* astérisque
4	O rond
5	X croix
≥ 6	réservé à l'enregistrement ou dépend du matériel

(*) Aspect Source Flag ASF (i)=0 référence à la table de groupage du poste
 ASF (i)=1 référence individuelle

⇨ TEXT

Tracé de texte

GKS

CALL GTX(X,Y,TEXTE)

PARAMETRES :

X	En, R	abscisse du point de départ du texte
Y	En, R	ordonnée
TEXTE	En, C	chaîne de caractères constituant le texte

EFFET :

- Une chaîne de caractères est générée à la position indiquée. La position du texte est donnée en coordonnées universelles (WC). Celle-ci subit la transformation de normalisation. Le texte sera découpé selon le facteur de précision du texte.
- Les valeurs courantes des attributs de texte sont attachées à la primitive.

ATTRIBUTS :

G NG		ASF(.)
X	Index de texte courant	CALL GSTXI(INDEX)
X	Police et précision	7 CALL GSTXFP(IPOL,IPREC)
X	Sens de l'écriture	CALL GSTXP(IDIREC)
X	Alignement	CALL GSTXAL(XAL, YAL)
X	Facteur d'agrandissement	8 CALL GSCHXP(ETEND)
X	Espacement	9 CALL GSCHSP(ESPA)
X	Hauteur des caractères	CALL GSCHH(TAILLE)
X	Orientation	CALL GSCHUP(XUP,YUP)
X	Index de couleur	10 CALL GSTXCI(ICOUL)

CALL GSTXR(IDPT,INDEX,IPOL,IPREC,ETEND,ESPA,ICOUL)

Sens de l'écriture	Alignement horizontal	Alignement vertical
0 droite	0 normal	0 normal
1 gauche	1 gauche	1 sommet
2 haut	2 centre	2 crête
3 bas	3 droite	3 mi-hauteur
		4 base
		5 bas

 **FILL AREA**

Tracé de polygone



CALL GFA(NS,XS,YS)

PARAMETRES :

NS	En, I	nombre de points
XS	En, R	abscisses des sommets
YS	En, R	ordonnées des sommets

EFFET : - Un polygone est généré. Les valeurs courantes des attributs de polygone sont attachées à la primitive.

ATTRIBUTS (non géométriques) :

-	(*) ASF(.)		
Index de remplissage courant		CALL GSFAI(INDEX)	En, I
Style de surface	11	CALL GSFAIS(ISTYLS)	En, I
Style intérieur	12	CALL GSFASI(ISTYLI)	En, I
Index de couleur	13	CALL GSFACI(ICOUL)	En, I

CALL GSFAR(IDPT,INDEX,ISTYLS,ISTYLI,ICOUL)

Le style de surface est défini comme suit :

0	contour (hollow)
1	plein (solid)
2	motif (pattern)
3	hachure (hatch)

Le style intérieur détermine la façon dont on remplit l'intérieur du polygone. Si l'attribut style de surface est contour ou plein, l'attribut style intérieur n'est pas utilisé. Par contre si le style de surface est motif ou hachure, alors l'attribut style intérieur précise le type de motif ou de hachure utilisé.

(*) Aspect Source Flag

ASF (i)=0
ASF (i)=1

référence à la table de groupage
référence individuelle

ATTRIBUTS (géométriques) :

- La taille du motif
- La position du point de référence

Ces deux attributs sont déterminés par une boîte de motif. La boîte de motif est un parallélogramme défini par les vecteurs LARGEUR DE MOTIF et HAUTEUR DE MOTIF positionnés par rapport à un POINT DE REFERENCE DU MOTIF. La boîte est divisée suivant une grille de NX x NY cellules.

Trois primitives permettent de modifier le MOTIF :

- | | |
|--------|---------------------------------------------|
| GSPARF | Définit le point de référence du motif (WC) |
| GSPA | Définit la taille du motif (WC) |
| GSPAR | Définit le motif |

⇒ **CREATE SEGMENT** Création d'un segment

CALL GCRSEG (NUMSEG)

PARAMETRE :

NUMSEG En, I numéro (nom) du segment à créer

EFFET :

- GKS est mis dans l'état opératoire SGOP. La liste d'état des segments est créée et initialisée. Le nom du segment est enregistré comme "nom du segment ouvert" dans la liste d'état de GKS.
- Toutes les primitives graphiques suivantes jusqu'à la prochaine fonction CLOSE SEGMENT seront rangées dans ce segment.
- Tous les postes de travail actifs sont inclus dans l'entrée "ensemble des postes de travail associés" de la liste d'état du segment.

⇒ **CLOSE SEGMENT** Fermeture du segment en cours



CALL GCLSG

EFFET :

- Le segment en cours est fermé.
- GKS retourne à l'état WSAC.
- L'entrée "nom du segment ouvert" de la liste d'état de GKS ne peut plus être interrogée.

⇒ **RENAME SEGMENT** Changer le nom d'un segment

CALL GRENSG(NUMSGA,NUMSGN)

PARAMETRES :

NUMSGA En, I ancien nom de segment
NUMSGN En, I nouveau nom de segment

EFFET :

- Chaque apparition de l'ancien nom de segment dans les tables de GKS est remplacée par le nouveau nom.

NOTE :

- L'ancien nom redevient utilisable

⇨ **DELETE SEGMENT** Destruction d'un segment



CALL GDSG(NUMSEG)

EFFET :

- Le segment est supprimé. Pour un poste Output ou Input, le segment est, si l'unité de visualisation est capable de réaliser l'opération, effacé de la surface d'affichage.

⇨ **DELETE SEGMENT FROM WORKSTATION**



Destruction d'un segment d'un poste de travail

CALL GDSGWK(IDPT,NUMSEG)

EFFET :

- Le segment est supprimé du poste de travail spécifié. Le nom du segment est retiré de la liste des segments stockés pour ce poste. L'identificateur du poste est retiré de la liste des postes associée à ce segment. Si cette liste devient vide le segment est alors totalement supprimé.

⇨ **ASSOCIATE SEGMENT WITH WORKSTATION**

Transmission d'un segment du WISS à un WDSS

CALL GASGWK(IDPT,NUMSEG)

EFFET :

- Le segment spécifié est transmis au poste de travail comme si ce dernier était actif lors de la création du segment. Les rectangles de découpage sont copiés tels quels.

NOTE :

- Le segment doit être dans le WISS.

⇨ COPY SEGMENT TO WORKSTATION

Transmission d'un segment du WISS à un poste de travail

CALL GCSGWK(IDPT,NUMSEG)

EFFET : - Les primitives du segment sont transmises au poste de travail après transformation du segment et découpage suivant le rectangle de découpage stocké avec chaque primitive. Elles ne sont pas stockées dans un segment.

⇨ INSERT SEGMENT Insertion de segment

CALL GINSG(NUMSEG,XMATI)

PARAMETRES :

NUMSEG En, I nom de segment à traiter
XMATI En, 2x3 R matrice de transformation d'insertion du segment

EFFET : - Les primitives de sortie (avec les valeurs de leurs attributs) sont extraites du poste WISS, puis subissent la transformation de segment. Après quoi GKS leur applique la transformation d'insertion spécifiée à l'appel. Cette transformation est caractérisée par une matrice 2 X 3.

NOTE : - Si le segment spécifié n'est pas dans le WISS ou est le segment ouvert, une erreur apparaît.
- Toutes les primitives conservent les valeurs des attributs qui leur ont été affectées lorsqu'elles ont été créées. La table de GKS n'est pas affectée par l'insertion.

⇨ SET SEGMENT TRANSFORMATION

Définition de la transformation de segment

CALL GSSGT(NUMSEG,XMATT)

PARAMETRES :

NUMSEG En, I numéro du segment
XMATT En, 2x3 R matrice de transformation du segment

EFFET : - L'entrée "Matrice de transformation de segment" dans la liste d'état des segments est mise à la valeur spécifiée.

⇨ EVALUATE TRANSFORMATION MATRIX

Calcul de la matrice de transformation

CALL GEVTM(PRX,PRY,TX,TY,TETA,SX,SY,INDCOR,XMAT)

PARAMETRES :

PRX	En, R	Abscisse du point de référence de la transformation
PRY	En, R	Ordonnée du point de référence de la transformation
TX	En, R	Coordonnée du vecteur de translation suivant x
TY	En, R	Coordonnée du vecteur de translation suivant y
TETA	En, R	Angle de rotation (radian)
SX	En, R	Facteur d'échelle suivant x
SY	En, R	Facteur d'échelle suivant y
INDCOR	En, I	Index de type de coordonnées(O : WC, 1 : NDC)
XMAT	So, 2x3 R	Matrice de transformation évaluée

EFFET :

- Si l'indicateur indique l'espace WC, les coordonnées du point de référence (point fixe) et celle du vecteur de translation subissent aux préalable la transformation de normalisation.
- L'ordre de transformation est mise à l'échelle, rotation, translation. Le point fixe est commun aux deux premières applications.
- La matrice produite pourra être utilisée par les fonctions SET SEGMENT TRANSF et INSERT SEGMENT.

⇨ ACCUMULATE TRANFROMATION MATRIX

Accumulation de matrices de transformation

CALL GACTM(XMAT1,PRX,PRY,TX,TY,TETA,SX,SY,INDCOR,XMAT2)

EFFET :

- La matrice XMAT2 contient la composée de la matrice XMAT1 avec la transformation calculée avec les paramètres d'entrée.

⇨ SET VISIBILITY Visibilité / invisibilité 60

CALL GSVIS(NUMSEG,IVIS)

PARAMETRES :

NUMSEG	En, I	nom du segment
IVIS	En, I	indicateur de visibilité

Effet :

- Le segment spécifié est rendu visible (IVIS=1) ou invisible (IVIS=0). Par défaut (IVIS=1). Un segment rendu invisible devient indétectable ; s'il est affiché, il sera effacé si le poste de travail est capable de réaliser l'opération.

⇨ SET HIGHLIGHTING Mise en valeur



CALL GSPLIT(NUMSEG,IBRIL)

PARAMETRES :

NUMSEG	En, I	nom du segment
IBRIL	En, I	indicateur de mise en évidence

EFFET :

- Si le segment est visible et suivant les possibilités du poste de travail le segment désigné est mis en évidence

IBRIL = 0 affichage normal
IBRIL = 1 affichage en surbrillance

⇨ SET SEGMENT PRIORITY Priorité d'un segment

CALL GSSGP(NUMSEG,PRIO)

PARAMETRES :

NUMSEG	En, I	nom du segment
PRIO	En, R	priorité du segment [0., 1.]

EFFET :

- Dans le cas d'un réaffichage de segments, l'ordre d'apparition des primitives va du moins au plus prioritaire, (Pb de taille entre objet). Pour une désignation d'un point commun à plusieurs segments, ce sera le segment le plus prioritaire qui sera retenu comme contenant ce point.

⇒ **SET DETECTABILITY** DéTECTABilité d'un segment ↗

CALL GSDTEC(NUMSEG,IDEPEC)

PARAMETRES :

NUMSEG En, I nom du segment
IDEPEC En, I indicateur de détectabilité

EFFET : - Pour pouvoir être désigné un segment doit être détectable. Il doit être en outre également visible.

IDEPEC = 0 segment non détectable
IDEPEC = 1 segment détectable

NOTE : - Le segment spécifié peut être ouvert ou fermé.

⇨ WRITE ITEM TO GKSM

Ecriture d'enregistrement dans le métafichier

CALL GWITM(IDPTMO,ISORT,IDL,INR,BUF)

PARAMETRES :

IDPTMO	En, I	identificateur du poste de travail
ISORT	En, I	type de l'enregistrement
IDL	En, I	nombre de caractères de l'enregistrement(*)
INR	En, I	nombre d'articles dans l'enregistrement (n)
BUF	En, nC80	données de l'enregistrement

(*) pas utilisé dans GKSBy

EFFET : - Ecrit dans un métafichier un enregistrement de données **non-graphiques**. Les données sont transmises automatiquement dès qu'un poste de travail de la catégorie sortie sur (MO) a été activé.

⇨ GET ITEM TYPE FROM GKSM

Obtention du type de l'item

CALL GGTITM(IDPTMI,TYPITM,LITMEM)

PARAMETRES :

IDPTMI	En, I	identification du poste de travail (Metafile input)
TYPITM	So, I	type de l'enregistrement
LITEM	So, I	longueur en nombre de caractères de l'article

EFFET : - Obtenir le type de l'enregistrement courant et la longueur de l'enregistrement.

⇨ READ ITEM FROM GKSM

Lecture d'enregistrement dans un métafichier

CALL GRDITM(IDTPMI,LMAX,INR,BUF)

PARAMETRES :

IDPTMI	En, I	identificateur du poste de travail
LMAX	En, I	taille maximum de l'enregistrement de données (car.)
INR	So, I	taille actuelle de l'enregistrement (n articles de 80)
BUF	So, nC80	enregistrement de données

- EFFET :**
- Lit dans le métafichier (MI) l'enregistrement courant et le retourne dans le tampon.
 - L'enregistrement suivant devient l'enregistrement courant.
 - Si (LMAX=0) l'enregistrement est ignoré.

⇨ INTERPRET ITEM

Interprétation d'article

CALL GIITM(TYPITM,LITM,INR,BUFI)

PARAMETRES :

TYPITM	En, I	type de l'enregistrement
LITM	En, I	longueur de l'enregistrement de données (car) (*)
INR	En, I	longueur de l'enregistrement (n articles de 80)
BUF	En, nC80	enregistrement de données

(*) pas utilisé dans GKS_{Bx}

- EFFET :**
- L'enregistrement fourni est interprété. Les variables d'état de GKS sont modifiées et les primitives graphiques sont générées.

⇨ **INITIALISE LOCATOR** Initialization du dispositif de Désignation

**CALL GINLC(IDENT,NRC,INT,RPX,RPY,ITEN,RMINX,
RMAXX,RMINY,RMAXY,ITAIL,DATREC)**

PARAMETRES :

IDENT	En, I	identificateur du poste de travail
NRC	En, I	numéro du releveur de coordonnées
INT	En, I	numéro initial de transformation de normalisation
RPX,RPY	En, R	position initiale du dispositif de désignation en (WC)
ITEN	En, I	types d'écho / notifications
RMINX,		
RMAXX,		
RMINY,		
RMAXY	En, R	zone d'écho en coordonnées (DC).
ITAIL	En, I	taille de l'enregistrement de données
DATREC	En, C	enregistrement de données

EFFECT :

(RPX,RPY), INT,ITEN,RMINX,RMAXX,...sont stockés dans les entrées correspondantes de la table d'état du poste de travail concernant le releveur de coordonnées spécifié. Pour certains types d'écho / notifications de releveur de coordonnées, 2 positions sont requises. Une des positions qui demeure fixe pendant l'opération d'entrée est la position "initiale" du releveur de coordonnées. L'autre position est la position "courante" du releveur de coordonnées. Elle varie dynamiquement lorsque l'opérateur utilise le releveur de coordonnées.

Types d'écho / notification

1 Désigne la position courante du releveur de coordonnées au moyen d'une technique définie par la réalisation.

2 Le réticule désigne la position courante du releveur de coordonnées à l'aide d'une ligne horizontale et d'une ligne verticale présentées sur la surface d'affichage, ou sur la clôture du poste de travail. L'intersection de ces deux lignes définit la position courante du releveur de coordonnées.

3 Désigne la position courante du dispositif de désignation à l'aide d'une croix de poursuite.

4 Désigne la position courante du releveur de coordonnées à l'aide d'un élastique reliant la position initiale du releveur de coordonnées fournie par cette fonction et la position courante.

5 Désigne la position courante du releveur de coordonnées à l'aide d'un rectangle. La diagonale du rectangle est la ligne reliant la position initiale du releveur de coordonnées et la position courante de ce relevé.

6 Affiche une représentation digitale de la position courante du releveur de coordonnées en coordonnées dépendant de l'appareil de relevé, dans la zone d'écho.

> 7 Les notifications et échos dépendent de l'appareil de relevé.

⇨ SET LOCATOR MODE

Initialisation du mode du dispositif de désignation (locator)

CALL GSLCM (IDENT, LCDNR, MODE, ESW)

PARAMETRES :

IDENT	En, I : identificateur du poste de travail
LCDNR	En, I : numéro du releveur de coordonnées
MODE	En, I : mode opératoire 0 : requête 1 : échantillonnage 2 : événement
ESW	En, I : mode commande d'écho 0 : pas d'écho 1 : écho

EFFECT :

Le releveur de coordonnées donné est mis au mode opératoire spécifié et son état d'écho est mis ECHO 1 ou PAS ECHO 0. Selon le mode opératoire spécifié, une interaction avec l'appareil donné peut commencer ou se terminer. L'état de l'appareil d'entrée défini par "MODE OPERATOIRE" et "COMMANDE D'ECHO" est mémorisé dans l'entrée concernant le releveur de coordonnées spécifié de la table d'état du poste de travail.

⇨ REQUEST LOCATOR

Localisation d'un point

CALL GRQLC(IDPT,MATER,ISTAT,NUM,XP,YP)

PARAMETRES :

IDPT En, I : identificateur du poste de travail
MATER En, I : matériel commandant la localisation

Exemple :

1 par clavier
2 par tablette graphique

ISTAT So, I : statut
 0 : localisation annulée
 1 : un point a été localisé
NUM So, I : numéro de la transformation normalisée
XP,YP So, R : coordonnées (WC) du point localisé

EFFECT :

Cette fonction permet de localiser un point par déplacement d'un réticule, à l'aide : soit du clavier (flèches,roulettes,...), soit d'une tablette graphique (tablette à digitaliser). Lorsque le réticule est positionné, taper un caractère (par ex : "espace") pour valider la position ; les coordonnées du point courant sont alors renvoyées dans les paramètres XP et YP.

⇨ INITIALISE STROKE**Initialisation du releveur d'une suite de coordonnées****CALL GINSK(IDENT,NRC,INT,N,RPX,RPY,ITEN,RMINX,
RMAXX,RMINY,RMAXY,IBUF,ITAIL,DATREC)****PARAMETRES :**

IDENT	En, I :	identificateur du poste de travail
NRC	En, I :	numéro d'appareil logique d'entrée
INT	En, I :	numéro de la transformation de normalisation initiale
N	En, I :	nombre de points de la courbe initiale
RPX,RPY	En, R :	points de la courbe initiale en coordonnées (WC)
ITEN	En, I :	type d'écho / notification
RMINX,		
RMAXX,		
RMINY,		
RMAXY	En, R :	zone d'écho en coordonnées (DC)
IBUF	En, I :	taille du tampon d'entrée pour le releveur
ITAIL	En, I :	taille de l'enregistrement de données
DATREC	En, C :	enregistrement de données

EFFET :

RPX(N), RPY(N), INT, ITEN, RMINX, RMAXX, RMINY, RMAXY, DATREC sont mémorisés dans l'entrée, concernant le releveur d'une suite de coordonnées spécifiées de la table d'état du poste de travail.

Pour tous les types d'écho / notification, la première valeur de l'enregistrement est la taille du tampon d'entrée, exprimée en entier ; la taille est comparée à une taille maximum de tampon définie par la réalisation pour ce type d'appareil.

Si la taille demandée du tampon est supérieure à cette taille maximum, elle est remplacée dans l'enregistrement par la taille maximum d'un tampon d'entrée défini par la réalisation. Si la courbe initiale déborde du tampon, il se produit une erreur. Quand le processus de mesure de releveur d'une suite de coordonnées est lancé, il est doté d'un tampon ayant pour taille, la taille courante du tampon d'entrée. La courbe initiale est copiée dans le tampon et la position d'édition est la position d'édition initiale du tampon défini dans l'enregistrement. La modification des points commence à cette position si elle n'a pas été définie dans l'enregistrement, la valeur utilisée est 1.

Type d'écho / notification

- 1 Affiche la courbe courante en utilisant une technique définie par la réalisation.
- 2 Affiche une représentation digitale de la position courante de la courbe à l'intérieur de la zone d'écho.
- 3 Affiche une marque à chaque point de la courbe courante.
- 4 Affiche une ligne joignant consécutivement les points de la courbe courante.
- > 5 Echos et notifications dépendant de l'appareil relevage d'une suite de coordonnées.

NOTES :

Pour tout type d'écho / notification, l'enregistrement concernant le relevage d'une suite de coordonnées peut contenir une position initiale d'édition pour le contenu du tampon dont la valeur peut varier de 1 au nombre de points +1 de la courbe.

⇨ SET STROKE MODE

Initialisation du mode du dispositif de désignation (stroke)

CALL GSSKM (IDENT,SKDNR,MODE,ESW)

PARAMETRES :

IDENT	En, I :	identificateur du poste de travail
SKDNR	En, I :	numéro du relevage d'une suite de coordonnées.
MODE	En, I :	mode opératoire 0 : requête 1 : échantillonnage 2 : événement
ESW	En, I :	mode commande d'écho 0 : pas d'écho 1 : écho

EFFET :

Le relevage de coordonnées donné est mis au mode opératoire spécifié et son état d'écho est mis ECHO 1 ou PAS ECHO 0. Selon le mode opératoire spécifié, une interaction avec l'appareil donne peut commencer ou se terminer. L'état de l'appareil d'entrée défini par "MODE OPERATOIRE" et "COMMANDE D'ECHO" est mémorisé dans l'entrée concernant le relevage de coordonnées spécifié de la table d'état du poste de travail.

⇨ REQUEST STROKE Entrée d'une suite de coordonnées**CALL GRQSK(IDENT,NREL,N,IETAT,NUMT,NP,RX,RY)****PARAMETRES :**

IDENT	En, I :	identificateur du poste de travail
NREL	En, I :	numéro du releveur d'une suite de coordonnées
N	En, I :	nombre maximum de points (n)
IETAT	So, I :	Etat 0 : Break 1 : OK
NUMT	So, I :	numéro de la transformation de normalisation
NP	So, I :	nombre de points
RX,RY	So, n R :	coordonnées (WC) des points de la courbe

EFFET :

GKS effectue une requête sur le releveur d'une suite de coordonnées. Si la fonction de rupture est appelée par l'opérateur l'état renvoyé (IETAT) prend la valeur 0 (Break), sinon l'état 1 (OK) est renvoyé avec la valeur logique d'entrée qui est la mesure courante du releveur d'une suite de coordonnées. Cette mesure est composée d'une série de points en coordonnées universelles et d'un numéro de transformation de normalisation qui a été utilisée pour effectuer la conversion en coordonnées universelles. Les points de la courbe doivent tous être à l'intérieur de la fenêtre associée à la transformation de normalisation.

NOTE :

Si l'opérateur rentre plus de points que ne le permet le tampon d'entrée (dans la liste des stations) les points additionnels sont perdus.

⇨ **INITIALISE VALUATOR** Initialisation de l'appareil
d'acquisition de valeur

CALL GINVL(IDENT,NAC,IV,ITN,XMIN,XMAX,YMIN,YMAX,
RMINV,RMAXV,ITAIL,DATREC)

PARAMETRES :

IDENT	En, I :	identificateur du poste de travail
NAC	En, I :	numéro de l'appareil comparateur
IV	En, I :	valeur initiale
ITN	En, I :	type d'écho / notification
XMIN,		
XMAX,		
YMIN,		
YMAX	En, R :	zone d'écho en coordonnées (DC)
RMINV	En, R :	valeur minimale
RMAXV	En, R :	valeur maximale
ITAIL	En, I :	taille de l'enregistrement de données
DATREC	En, C :	enregistrement de données

EFFET :

La valeur initiale, le type d'écho / notification, la zone d'écho et l'enregistrement de données sont stockés dans les entrées correspondantes à l'entrée scalaire spécifiée dans la table d'état du poste de travail. Pour tous les types d'écho / notification d'entrée scalaire, l'enregistrement de données devrait contenir une valeur basse et une valeur haute spécifiant l'intervalle. Les valeurs provenant de l'appareil seront rapportées à cet intervalle par un changement d'échelle linéaire.

Type d'écho / notification

- 1 Désigne la valeur courante de l'entrée scalaire au moyen d'une technique définie par la réalisation.
- 2 Affiche une représentation graphique de la valeur courante de l'entrée scalaire dans la zone d'écho (par exemple un cadran ou une flèche).
- 3 Affiche une représentation digitale de la valeur courante de l'entrée scalaire dans la zone d'écho.
- > 4 Les notifications et les échos dépendent de l'entrée scalaire.

⇒ SET VALUATOR MODE

initialisation du mode du dispositif de désignation (valuator)

CALL GSVLM(IDENT,VLDNR,MODE,ESW)

PARAMETRES :

IDENT	En, I :	identificateur du poste de travail
VLDNR	En, I :	numéro du releveur d'entrée scalaire
MODE	En, I :	mode opératoire 0 : requête 1 : échantillonnage 2 : événement
ESW	En, I :	mode commande d'écho 0 : pas d'écho 1 : écho

EFFET :

Le releveur d'entrée scalaire donné est mis au mode opératoire spécifié et son état d'écho est mis ECHO 1 ou PAS ECHO 0. Selon le mode opératoire spécifié, une interaction avec l'appareil donné peut commencer ou se terminer. L'état de l'appareil d'entrée défini par "MODE OPERATOIRE" et "COMMANDE D'ECHO" est mémorisé dans l'entrée concernant le releveur de coordonnées spécifié de la table d'état du poste de travail.

⇒ REQUEST VALUATOR Requête de scalaire

CALL GRQVL(IDENT,NUM,IETAT,RVAL)

PARAMETRES :

IDENT	En, I :	identificateur du poste de travail
NUM	En, I :	numéro de l'entrée de scalaire
IETAT	So, I :	état 0 : Break 1 : OK
RVAL	So, R :	valeur logique d'entrée

EFFET :

GKS effectue une requête sur l'entrée de scalaire spécifiée. Si la fonction de rupture est appelée par l'opérateur, l'état renvoyé (IETAT) prend la valeur 0 (Break), sinon l'état 1 (OK) est renvoyé avec la valeur logique d'entrée qui est la mesure courante de l'entrée de scalaire. La valeur fournie est dans l'intervalle spécifié dans l'entrée de la table d'état du poste de travail (dans l'enregistrement de données).

⇨ INITIALISE CHOICE

Initialisation de sélection

**CALL GINCH (IDENT,ISEL,ISTAT,INITI,IECHO,RXMIN,
RXMAX,RYMIN,RYMAX,ILONG,DATREC)**

PARAMETRES :

IDENT	En, I :	identificateur du poste de travail
ISEL	En, I :	numéro de l'appareil sélecteur
ISTAT	En, I :	statut initial (GOK, GNCHOI)
INITI	En, I :	numéro initial du sélecteur
IECHO	En, I :	type d'écho / notification
RXMIN,		
RMAXX,		
RMINY,		
RMAXY	En, R :	zone d'écho en coordonnées (DC)
ILONG	En, I :	longueur de l'enregistrement de données
DATREC	En, C :	enregistrement de données

EFFECT :

Le numéro initial du sélecteur, le type d'écho / notification, la zone d'écho et l'enregistrement de données sont rangés dans les entrées correspondantes concernant le sélecteur spécifié de la table d'état du poste de travail.

Type d'écho / notification

1 Désigne le numéro du sélecteur courant au moyen d'une technique définie par la réalisation.

2 Les appareils physiques d'entrée qui sont utilisés le plus couramment pour mettre en oeuvre un appareil logique d'entrée de sélecteur comportent normalement une fonction intégrée de notification. Ce type d'écho / notification permet au programme d'application d'appeler cette fonction de notification. Si la valeur du ième élément du "TABLEAU DE NOTIFICATIONS" de l'enregistrement de données est "OFF", la notification de la ième alternative du sélecteur d'entrée spécifié est impossible. Si cette valeur est "ON", la fonction de notification de cette alternative est utilisable. Le premier élément de l'enregistrement indique le nombre d'alternatives définies par la réalisation pour cet appareil. Si cette valeur maximum est dépassée, une erreur est produite. Le second élément de l'enregistrement est constitué par le tableau de notifications.

3 Permet à l'opérateur d'indiquer un numéro de sélection en sélectionnant au moyen d'une technique appropriée une chaîne parmi plusieurs chaînes de caractères. Les chaînes sont contenues dans l'enregistrement des données et sont affichées dans la zone d'écho. La valeur logique d'entrée est le numéro de la chaîne sélectionnée.

Le premier élément de l'enregistrement est le nombre de choix.

Remarques :

Avec GKS_{Bx}, ILONG représente le nombre de choix qui seront réalisés par l'application. (cf programme d'application avec GKS et les remarques sur GRQCH & GPREC & GUREC).

Il est comparé au nombre maximum d'alternative possibles définies à la réalisation pour cet appareil. Si cette valeur est dépassée une erreur est générée. Le 2ème élément de l'enregistrement est le " TABLEAU DES CHAINES OPTIONS".

4 Permet à l'opérateur d'indiquer un numéro de sélection en sélectionnant au moyen du clavier, une chaîne de caractère parmi plusieurs chaînes. Les chaînes sont contenues dans l'enregistrement de données et elles peuvent être affichées dans la zone d'écho sous la forme d'une notification. La chaîne rentrée par l'opérateur est répercutée en écho dans la zone d'écho. La valeur logique d'entrée est le numéro de la chaîne dont l'opérateur a effectué l'entrée. Le premier élément de l'enregistrement est le nombre de chaînes. Il est comparé au nombre maximum d'alternatives définies par la réalisation pour cet appareil. Si cette valeur maximum est dépassée, cela provoque une erreur. Le second élément de l'enregistrement est le "TABLEAU DES CHAINES OPTIONS".

5 Le segment nommé par l'enregistrement de données est interprété pendant l'exécution de la fonction INITIALISATION DE SELECTEUR (GINCH) pour l'emploi ultérieur comme notification de l'appareil sélecteur spécifié. Il sera affiché dans la zone d'écho en transformant le carré unitaire [0,1] x [0,1] de l'espace des coordonnées sur la zone d'écho. Les identificateurs de désignation du segment sont transposés en valeurs numériques suivant des modalités dépendant du sélecteur.

La désignation de ces primitives définit la valeur correspondante. Après l'interprétation, il n'existe plus de connexion possible entre le segment et le sélecteur spécifiés. L'élément spécifié de l'enregistrement est le nom du segment.

> 6 Les notifications et l'écho dépendent du sélecteur.

⇨ **SET CHOICE MODE** Initialisation du mode du dispositif de désignation (choice)

CALL GSCHM(IDENT,CHDNR,MODE,ESW)

PARAMETRES :

IDENT	En, I :	identificateur du poste de travail
CHDNR	En, I :	numéro du sélecteur
MODE	En, I :	mode opératoire 0 : requête 1 : échantillonnage 2 : événement
ESW	En, I :	mode commande d'écho 0 : pas d'écho 1 : écho

EFFECT :

Le sélecteur donné est mis au mode opératoire spécifié et son état d'écho est mis ECHO 1 ou PAS ECHO 0. Selon le mode opératoire spécifié, une interaction avec l'appareil donné peut commencer ou se terminer. L'état de l'appareil d'entrée défini par "MODE OPERATOIRE" et "COMMANDE D'ECHO" est mémorisé dans l'entrée concernant le relevage de coordonnées spécifié de la table d'état du poste de travail.

⇨ REQUEST CHOICE Requête de choix

CALL GRQCH(IDPT,MATER,ISTAT,ICHOIX)

PARAMETRES :

IDPT En, I : identificateur du poste de travail
MATER En, I : matériel commandant le choix
 1 : clavier

ISTAT So, I : statut
 0 : choix annulé
 1 : choix effectué
 2 : pas de choix (choix= 0)
ICHOIX So, I : choix effectué

EFFECT :

Cette fonction permet d'effectuer un choix dans un menu, par le clavier.

⇨ INITIALISE PICK

Initialisation de l'identificateur de désignation

CALL GINPK(IDENT,NRC,INET,INSEG,INDES,ITEN,RMINX,
RMAXX,RMINY,RMAXY,ITAIL,DATREC)

PARAMETRES :

IDENT	En, I :	identificateur du poste de travail
NRC	En, I :	numéro du releveur de coordonnées
INET	En, I :	état initial 0 Break 1 Designe 2 Rien
INSEG	En, I :	numéro initial du segment
INDES	En, I :	identificateur initial de désignation
ITEN	En, I :	types d'écho / notifications
RMINX,		
RMAXX,		
RMINY,		
RMAXY	En, R :	zone d'écho en coordonnées (DC)
ITAIL	En, I :	taille de l'enregistrement de données
DATREC	En, C :	enregistrement de données

EFFECT :

ITEN, RMINX, RMINY, RMAXX, RMAXY, INET, INSEG, INDES et DATREC sont mémorisés dans les entrées concernant le releveur de coordonnées spécifiés dans la table d'état du poste de travail.

Type d'écho / notification

- 1 Utilise une technique définie par la réalisation et qui met au moins en évidence la primitive "désignée" pendant une courte période de temps.
- 2 Provoque un écho du groupe contigu de primitives comportant le même identificateur de désignation que la primitive "désignée" ou toutes les primitives du segment ayant le même identificateur de désignation que la primitive "désignée".
- 3 Provoque un écho de tout le segment contenant la primitive "désignée".
- 4 La notification et l'écho dépendent du dispositif de désignation.

⇨ SET PICK MODE

Initialisation du mode du dispositif de désignation (pick)

CALL GSPKM(IDENT,PKDNR,MODE,ESW)

PARAMETRES :

IDENT	En,I :	identificateur du poste de travail
PKDNR	En,I :	numéro du dispositif de désignation
MODE	En,I :	mode opératoire 0 : requête 1 : échantillonnage 2 : événement
ESW	En,I :	mode commande d'écho 0 : pas d'écho 1 : écho

EFFECT :

Le dispositif de désignation donné est mis au mode opératoire spécifié et son état d'écho est mis ECHO 1 ou PAS ECHO 0. Selon le mode opératoire spécifié, une interaction avec l'appareil donné peut commencer ou se terminer. L'état de l'appareil d'entrée défini par "MODE OPERATOIRE" et "COMMANDE D'ECHO" est mémorisé dans l'entrée concernant le relevage de coordonnées spécifié de la table d'état du poste de travail.

⇨ REQUEST PICK Détection de segment

CALL GRQPK(IDPT,MATER,ISTAT,NUM,IPICK)

PARAMETRES :

IDPT	En, I :	identificateur du poste de travail
MATER	En, I :	matériel commandant la désignation
		Exemple : 1 clavier 2 par tablette graphique
ISTAT	So, I :	statut 0 : désignation annulée 1 : désignation effectuée 2 : échec de la désignation
NUM	So, I :	numéro du segment désigné
IPICK	So, I :	identificateur de désignation

EFFECT :

Cette fonction permet de désigner un segment affiché, avec à l'intérieur de ce segment une partie repérée par un identificateur de "PICK". Cet identificateur de "PICK" a été associé à une primitive de sortie à l'intérieur du segment, par la fonction GSPKID.

⇨ INITIALISE STRING

Initialisation du releveur de caractères

CALL GINST(IDENT,STDNR,ISTR,PET,XMIN,XMAX,YMIN,
YMAX,BUFLEN,INIPOS,LDR,DATREC)

PARAMETRES :

IDENT	En, I :	identificateur du poste de travail
STDNR	En, I :	numéro du releveur de chaîne de caractères
ISTR	En, C :	chaîne de caractères initiale
PET	En, I :	types d'écho / notification
XMIN,		
XMAX,		
YMIN,		
YMAX	En, R :	zone d'écho en coordonnées (DC)
BUFLEN	En, I :	taille du buffer d'entrée.
INIPOS	En, I :	position du curseur dans la chaîne initiale
LDR	En, I :	taille de l'enregistrement de données
DATREC	En, C :	enregistrement de données

EFFET :

ISTR, PET, XMIN, YMIN, XMIN, XMAX, BUFLEN, INIPOS, LDR, et DATREC sont mémorisés dans les entrées concernant le releveur de chaîne de caractères spécifié dans la table d'état du poste de travail.

Pour tous les types d'écho / notification, le premier élément de l'enregistrement est la taille du tampon d'entrée entier appartenant à l'intervalle [1, n]. Cette taille est comparée à la taille maximum du tampon d'entrée pour cet appareil définie par la réalisation (contenu dans la table de description du poste de travail). Si la taille demandée du tampon est plus grande, elle est remplacée dans l'enregistrement par la taille maximum du tampon d'entrée.

Si la chaîne initiale a une longueur supérieure à la taille du tampon, cela provoque une erreur. Pour tous les types d'écho / notification, le second élément de l'enregistrement correspond à une position initiale du curseur qui peut varier dans l'intervalle 1 à la longueur de la chaîne plus 1.

Lorsqu'un processus de mesure de chaîne de caractères est lancé, il est doté d'un tampon de la taille courante du tampon d'entrée. La chaîne initiale est copiée dans le tampon et le curseur est mis à la position initiale du curseur dans ce tampon.

Le remplacement des caractères commence à cette position du curseur.

Type d'écho / notification

- 1 Affiche la valeur courante de la chaîne dans la zone d'écho.
- >= 2 La notification et l'écho dépendent de l'appareil.

⇨ SET STRING MODE

Initialisation du mode du dispositif de désignation (string)

CALL GSSTM(IDENT,STDNR,MODE,ESW)

PARAMETRES :

IDENT	En, I:	identificateur du poste de travail
STDNR	En, I:	numéro du releveur de chaîne de caractères
MODE	En, I:	mode opératoire 0 : requête 1 : échantillonnage 2 : événement
ESW	En, I:	mode commande d'écho 0 : pas d'écho 1 : écho

EFFECT :

Le releveur de chaîne de caractères donné est mis au mode opératoire spécifié et son état d'écho est mis ECHO 1 PAS ECHO 0. Selon le mode opératoire spécifié, une interaction avec l'appareil donné peut commencer ou se terminer. L'état de l'appareil d'entrée défini par "MODE OPERATOIRE" et "COMMANDE D'ECHO" est mémorisé dans l'entrée concernant le releveur de coordonnées spécifié de la table d'état du poste de travail.

⇒ REQUEST STRING Entrée d'une chaîne de caractères**CALL GRQST(IDPT,MATER,ISTAT,NBCAR,TEXTE)****PARAMETRES :**

IDPT	En, I:	identificateur du poste de travail
MATER	En, I:	matériel commandant l'entrée 1 : par clavier
ISTAT	So, I:	statut 0 : entrée annulée 1 : une chaîne de caractères a été entrée
NBCAR	So, I:	nombre de caractères entrés
TEXTE	So, C:	chaîne entrée

EFFET :

Cette fonction permet d'introduire une chaîne de caractères par le clavier.

⇒ **GQDSP** Dimensions maximales de la surface d'affichage

CALL GQDSP(ITYPT,IERR,IUNIT,RLARGM,RHAUTM,RLARGR,RHAUTR)

PARAMETRES :

ITYPT	En, I :	type du poste de travail
IERR	So, I :	indicateur d'erreur.
IUNIT	So, I :	type d'unité 0 : mètres ; 1 : entier
RLARGM	So, R :	largeur maximum de la surface d'affichage (valeur représentant des mètres)
RHAUTM	So, R :	hauteur maximum de la surface d'affichage (valeur représentant des mètres)
RLARGR	So, I :	largeur maximum de la surface en unités raster
RHAUTR	So, I :	hauteur maximum de la surface en unités raster

EFFET :

Si l'information demandée est disponible, l'indicateur d'erreur est mis à 0 et les informations sont fournies dans les différentes variables de sortie.

⇒ **GQCNTN** Numéro courant de transformation de normalisation

CALL GQCNTN(IERR,NUM)

PARAMETRES :

IERR	So, I :	indicateur d'erreur
NUM	So, I :	numéro courant de transformation de normalisation

EFFET :

Si l'information est disponible, l'indicateur d'erreur est mis à 0, et une valeur est renvoyée dans la variable de sortie NUM.

⇨ GQNT

Transformation normalisée

CALL GQNT(NUM,IERR,WIN,WIEW)

PARAMETRES :

NUM	En, I : numéro de la transformation de normalisation
IERR	So, I : indicateur d'erreur
WIN	So, R4: limites de la fenêtre en coordonnées (WC) WIN(1) --> MIN sur axe X WIN(2) --> MAX sur axe X WIN(3) --> MIN sur axe Y WIN(4) --> MAX sur axe Y
VIEW	So, R4: limites de la clôture en coordonnées (NDC) VIEW(1) --> MIN sur axe X VIEW(2) --> MAX sur axe X VIEW(3) --> MIN sur axe Y VIEW(4) --> MAX sur axe Y

EFFET :

Si l'information demandée est disponible, l'indicateur d'erreur est mis à 0, et les valeurs sont retournées dans des différents paramètres.

⇨ GQCLIP Indicateur de découpage

CALL GQCLIP(INDERR,INDECR)

PARAMETRES :

INDERR	So, I : indicateur d'erreur
INDECR	So, I : indicateur de découpage 0 : pas de découpage 1 : découpage

EFFET :

Si l'information demandée est disponible, l'indicateur d'erreur est renvoyé à 0, et la valeur courante de l'indicateur de découpage est renvoyée dans le paramètre INDECR.

⇒ **GQLN** Type de ligne courante

CALL GQLN(INDERR,ITYPE)

PARAMETRES :

INDERR	So, I : indicateur d'erreur
ITYPE	So, I : type de ligne
	1 : continue
	2 : tiretée
	3 : pointillée
	4 : mixte

EFFET :

Si l'information demandée est disponible, l'indicateur d'erreur est renvoyé à 0, et la valeur courante du type de ligne est renvoyée dans le paramètre ITYPE.

⇒ **GQLWSC** Epaisseur courante de ligne

CALL GQLWSC(INDERR,WLARG)

PARAMETRES :

INDERR	So, I : indicateur d'erreur
WLARG	So, R : largeur de ligne

EFFET :

Si l'information demandée est disponible, l'indicateur d'erreur est renvoyé à 0, et la valeur courante de la largeur du polygone est renvoyée dans le paramètre WLARG.

⇒ **GQPLCI** Couleur courante de ligne

CALL GQPLCI (INDERR,ICOUL)

PARAMETRES :

INDERR So, I : indicateur d'erreur
ICOUL So, I : couleur de ligne

EFFET :

Si l'information demandée est disponible, l'indicateur d'erreur est renvoyé à 0, et la valeur courante de la couleur de ligne est renvoyée dans le paramètre ICOUL.

⇒ **GQMK** Type courant de marqueur

CALL GQMK(INDERR,ITYPE)

PARAMETRES :

INDERR So, I : indicateur d'erreur
ITYPE So, I : type de marque

EFFET :

Si l'information demandée est disponible, l'indicateur d'erreur est renvoyé à 0, et la valeur courante du type de marque est renvoyée dans le paramètre ITYPE.

⇒ **GQMKSC** Taille courante de marqueur

CALL GQMKSC(INDERR, TAILLE)

PARAMETRES :

INDERR So, I : indicateur d'erreur
TAILLE So, R : dimension du marque

EFFET :

Si l'information demandée est disponible, l'indicateur d'erreur est renvoyé à 0, et la valeur courante de la dimension de la marque est renvoyée dans le paramètre TAILLE.

⇒ **GQPMCI** Couleur courante de marqueur

CALL GQPMCI(INDERR, ICOUL)

PARAMETRES :

INDERR So, I : indicateur d'erreur
ICOUL So, I : couleur de la marque

EFFET :

Si l'information demandée est disponible, l'indicateur d'erreur est renvoyé à 0, et la valeur courante de la couleur de la marque est renvoyée dans le paramètre ICOUL.

⇨ **GQCHH** Hauteur des caractères

CALL GQCHH(INDERR, TAILLE)

PARAMETRES :

INDERR So, I : indicateur d'erreur
TAILLE So, R : hauteur de caractère

EFFET :

Si l'information demandée est disponible, l'indicateur d'erreur est renvoyé à 0, et la valeur courante de la hauteur de caractère est renvoyée dans le paramètre TAILLE.

⇨ **GQCHUP** Vecteur haut des caractères

CALL GQCHUP(INDERR, XVCAR, YVCAR)

PARAMETRES :

INDERR So, I : indicateur d'erreur
XVCAR,
YVCAR So, R : coordonnées du vecteur haut

EFFET :

Si l'information demandée est disponible, l'indicateur d'erreur est renvoyé à 0, et la valeur courante du vecteur haut de caractère est renvoyée dans les paramètres XVCAR et YVCAR.

⇨ GQTXP

Direction d'écriture du texte

CALL GQTXP (INDERR, IDIREC)

PARAMETRES :

INDERR	So, I : indicateur d'erreur
IDIREC	So, I : direction d'écriture du texte

0 : à droite
1 : à gauche
2 : en haut
3 : en bas

EFFET :

Si l'information demandée est disponible, l'indicateur d'erreur est renvoyé à 0, et la valeur courante de la direction d'écriture est renvoyée dans le paramètre IDIREC.

⇨ GQTXAL

Alignement du texte

CALL GQTXAL(INDERR, IALH, IALV)

PARAMETRES :

INDERR	So, I : indicateur d'erreur
IALH	So, I : alignement horizontal

0 : normal
1 : gauche
2 : centre
3 : droite

IALV	So, I : alignement vertical
------	-----------------------------

4 : normal
5 : tout en haut
6 : haut
7 : au milieu
8 : bas
9 : tout en bas

EFFET :

Si l'information demandée est disponible, l'indicateur d'erreur est renvoyé à 0, et la valeur courante de l'alignement (horizontal et vertical) est renvoyée dans les paramètres IALH et IALV.

⇒ **GQTXFP** Police et précision courante du texte

CALL GQTXFP(INDERR,IPOL,IPREC)

PARAMETRES :

INDERR	So, I : indicateur d'erreur
IPOL	So, I : police de caractères
IPREC	So, I : type de tracé 0 : chaîne 1 : caractère 2 : trait

EFFET :

Si l'information demandée est disponible, l'indicateur d'erreur est renvoyé à 0, et les valeurs courantes de la police et du type de tracé sont renvoyées dans les paramètres IPOL et IPREC.

⇒ **GQCHXP** Facteur d'expansion courant des caractères

CALL GQCHXP(INDERR,ETEND)

PARAMETRES :

INDERR	So, I : indicateur d'erreur
ETEND	So, R : facteur d'étendue des caractères

EFFET :

Si l'information demandée est disponible, l'indicateur d'erreur est renvoyé à 0, et la valeur courante de l'étendue des caractères est renvoyée dans le paramètre ETEND.

-⇒ **GQCHSP** Espacement entre caractères

CALL GQCHSP(INDERR,ESPA)

PARAMETRES :

INDERR So, I : indicateur d'erreur
ESPA So, R : espacement entre deux caractères

EFFECT :

Si l'information demandée est disponible, l'indicateur d'erreur est renvoyé à 0, et la valeur courante de l'espacement entre 2 des caractères est renvoyée dans le paramètre **ESPA**.

⇒ **GQTXCI** Couleur courante du texte

CALL GQTXCI(INDERR,ICOUL)

PARAMETRES :

INDERR So, I : indicateur d'erreur
ICOUL So, I : couleur du texte

EFFECT :

Si l'information demandée est disponible, l'indicateur d'erreur est renvoyé à 0, et la valeur courante de la couleur du texte est renvoyée dans le paramètre **ICOUL**.

⇒ **GQFAIS** Style courant de remplissage

CALL GQFAIS(INDERR,ISTYLE)

PARAMETRES :

INDERR So, I : indicateur d'erreur
ISTYLE So, I : style intérieur de la surface

EFFET :

Si l'information demandée est disponible, l'indicateur d'erreur est renvoyé à 0, et la valeur courante du style intérieur de la surface est renvoyée dans le paramètre ISTYLE.

⇒ **GQFASI** Index courant du style de remplissage

CALL GQFASI(INDERR,ISTYLE)

PARAMETRES :

INDERR So, I : indicateur d'erreur
ISTYLE So, I : index du style de la surface

EFFET :

Si l'information demandée est disponible, l'indicateur d'erreur est renvoyé à 0, et la valeur courante de l'index du style de la surface est renvoyée dans le paramètre ISTYLE.

⇒ **GQFACI** Couleur courant du remplissage de surface

CALL GQFACI(INDERR,ICOUL)

PARAMETRES :

INDERR So, I : indicateur d'erreur
ICOUL So, I : couleur du remplissage de surface

EFFET :

Si l'information demandée est disponible, l'indicateur d'erreur est renvoyé à 0, et la valeur courante de la couleur du remplissage est renvoyée dans le paramètre ICOUL.

⇒ GQASF Indicateurs de tracé**CALL GQASF(INDERR,ITBASF)****PARAMETRES :**

INDERR So, I : Indicateur d'erreur
ITBASF So, I : tableau des 13 indicateurs de tracé

ITBASF(i)=0 : Référence à la table de groupage
ITBASF(i)=1 : Référence individuelle

Signification des indicateurs :

- | | |
|----|-------------------------------------------|
| 1 | type de ligne |
| 2 | largeur de ligne |
| 3 | couleur de ligne |
| 4 | type de marque |
| 5 | dimension de la marque |
| 6 | couleur de la marque |
| 7 | précision du texte |
| 8 | facteur d'étendue des caractères |
| 9 | espacement entre deux caractères |
| 10 | couleur de texte |
| 11 | style intérieur de remplissage de surface |
| 12 | index de style de remplissage de surface |
| 13 | couleur de remplissage de surface |

EFFET :

Si l'information demandée est disponible, l'indicateur d'erreur est renvoyé à 0, et les valeurs courantes des indicateurs de tracé sont renvoyées dans le paramètre ITBASF.



TRAVAUX PRATIQUES

L'objectif est de réaliser, pendant cette session d'initiation à GKS, une application interactive offrant à l'utilisateur les fonctions élémentaires nécessaires à la production d'un objet graphique.

On choisit d'étudier et de réaliser un programme qui effectue le tracé d'une courbe prédéfinie. Cette application devra fonctionner de manière identique dans une fenêtre Suntools et sur un PC muni d'un interface EGA/VGA.

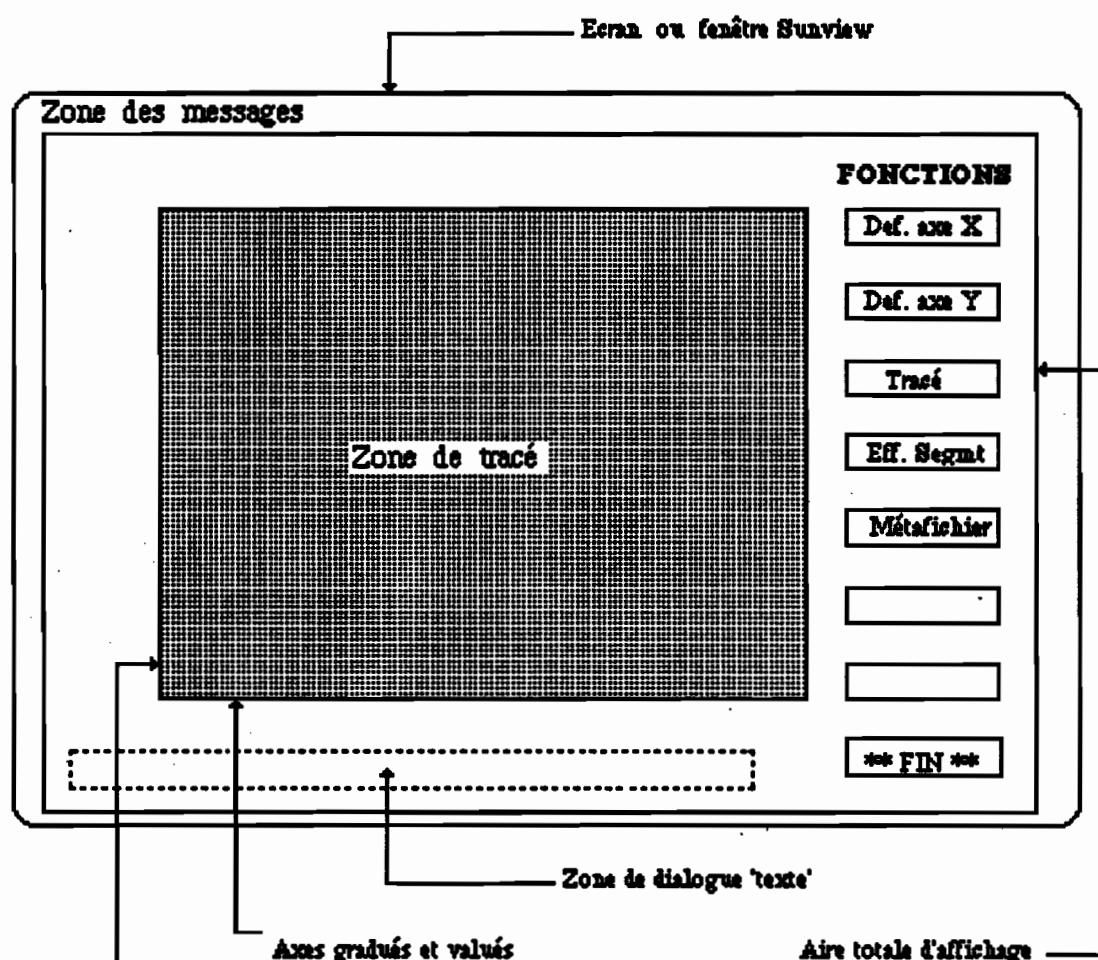
L'exercice est purement didactique. Il permet de mettre en œuvre les concepts fondamentaux de GKS. Le produit résultant pourra servir d'exemple à la réalisation d'une application professionnelle, par exemple le tracé de variables diverses sur un fond de carte.

Analyse de l'application :

Les fonctionnalités à mettre en œuvre dans cet exemple sont simples et ne nécessitent pas une attention particulière. Les fonctions élémentaires seront : le tracé d'un axe, le tracé d'un libellé, le tracé d'une courbe, l'effacement (redéfinition) d'un axe, la sauvegarde de l'objet graphique dans un métacharier.

L'aire d'affichage, (toute la fenêtre, tout l'écran), sera divisée en diverses zones : une zone de tracé, une zone de fonctions, une zone de dialogue (entrée de chaîne de caractères) et une zone de messages.

La figure ci-dessous représente l'aire d'affichage de l'application :



Cinq exercices permettent de réaliser l'application demandée. Chacun d'eux, à l'exception du premier, reprend le résultat du précédent. Dans chacune des étapes on tentera de dégager les fonctionnalités élémentaires réutilisables dans d'autres applications. Ainsi on développera, en parallèle de l'application proprement dite, une bibliothèque de modules de base.

Afin d'atteindre l'objectif fixé dans le délai imparti, une bibliothèque préexistente sera mise à la disposition des programmeurs afin de corriger leur propre programmation et d'assurer une certaine cohérence tout au long de cette session d'initiation.

Les modules de ce package sont des exemples, sans prétention, de sous-programmes réutilisables. Ils permettent notamment de gagner un temps précieux pendant la phase de développement. Cet ensemble de modules s'enrichira au fil du temps et devrait contenir à terme la quasi-totalité des outils de base à mettre en œuvre dans toute application GKS.

Comme dans toute bibliothèque, il sera bon de veiller à la cohérence, des modules entre eux, des définitions des structures de données, des appels de paramètres, bref de tout ce qui est à la base d'une programmation saine et portable.

Tous les textes source des divers exercices seront donnés aux participants. Les passages délicats seront commentés en cours.

Les sources des modules du package sont listés en annexe du polycopié à l'exception de ceux qui font office de TP.

EXERCICE 1

Le programme à rédiger sera nommé TPM1. Il devra réaliser les opérations suivantes:

⇒ **Ouverture de GKS**

Ecrire une routine opgks

subroutine opgks(luerr,ficerr)

luerr : unité logique du fichier des erreurs

ficerr: nom de ce fichier

Elle assure les fonctions suivantes : gestion des erreurs GKS, ouverture du système GKS.

⇒ **Ouverture et activation du poste de travail**

Ecrire une routine opwk

subroutine opwk(luwk,ficwk,idpt,itypt,ier,iqapa,numapa)

luwk : unité logique du fichier de sortie

ficwk : nom du fichier de sortie

idpt : identificateur du poste de travail

itypt : type de poste (driver)

ier : code d'erreur (entier)

iqapa : question num. appareil 1=oui

numapa: numéro d'appareil (1 par défaut)

Cette routine doit impérativement effectuer les opérations suivantes :

- entrée interactive du type de poste
- ouverture du poste et du fichier associé
- récupération des caractéristiques du poste
- activation du poste de travail
- définition de la clôture de la transformation de poste de travail en choisissant toute la surface de l'aire d'affichage
- calcul des ratios qui permettront de définir ultérieurement la fenêtre de la transformation de poste de travail (isométrie) et les clôtures des diverses transformations de normalisations nécessaires à l'application.

⇒ **Définition de la fenêtre de la transformation de poste de travail**

⇒ **Attente (Corps de l'application)**

⇒ **Désactivation du poste de travail**

⇒ **Fermerture du poste de travail**

⇒ **Fermerture de GKS**

Remarques:

Après toute exécution du programme on vérifiera le contenu du fichier des erreurs GKS, normalement aucun message ne doit apparaître. Ce fichier sera appelé une bonne fois pour toute gkserr.

Les identificateurs de poste de travail sont :

- 6971 poste de travail EGA/VGA sur PC
- 7281 poste de travail traceur HPGL (portrait)
- 7374 poste de travail monochrome sur SUN
- 7375 poste de travail couleur sur SUN

Le fichier de sortie associé au poste de travail sera nommé gkstra pendant tout le TP. La gestion de ce fichier doit être plus fine sous UNIX⁽¹⁾. Dans ce dernier cas il est conseillé d'utiliser une variable d'environnement, par exemple GKSFWK, pour spécifier le nom de ce fichier. La routine opwk ira tester si cette variable existe et au cas où récupérera son contenu par l'appel (call getenv(.....)). Cette technique s'avère très utile dans certains cas particuliers, par exemple quand on utilise une console compatible TEKTRONIX connectée au SUN. Dans ce cas on demandera au système le nom de la console avec la commande UNIX tty avant de lancer l'application. Si la réponse est /dev/ttya par exemple, on positionne la variable GKSFWK à cette valeur : setenv GKSFWK /dev/ttya.

(1) UNIX est une marque déposée des laboratoires Bell/ATT.

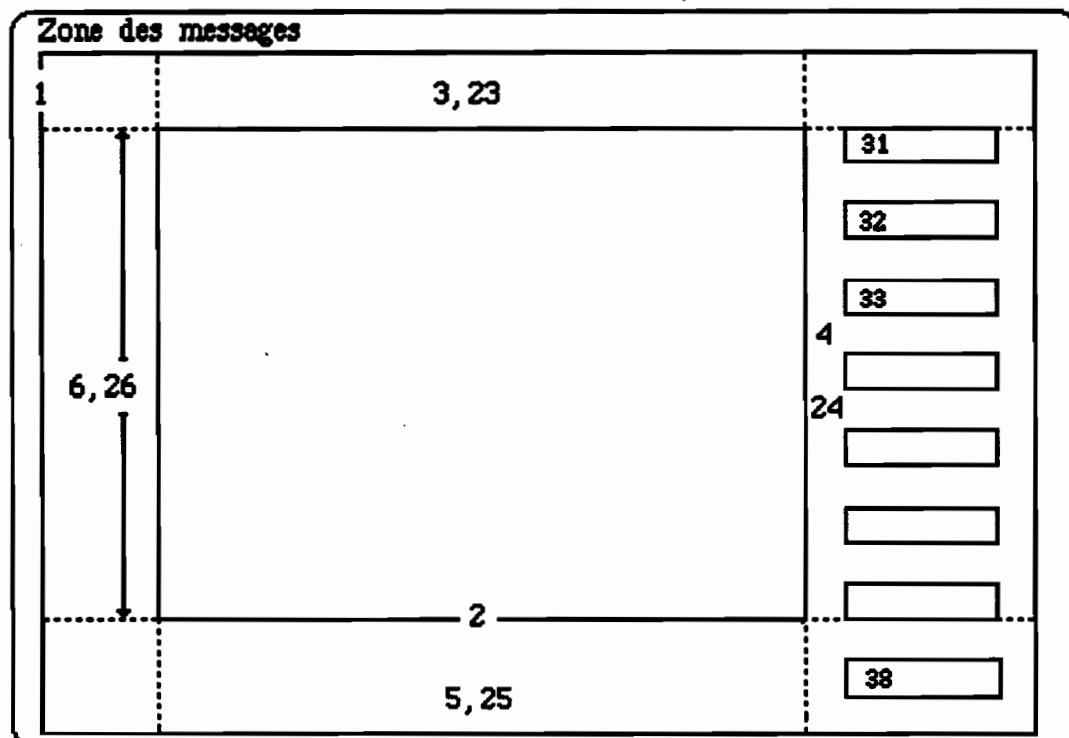
EXERCICE 2

Pour réaliser cet exercice on reprendra le texte de tpm1 que l'on recopiera pour produire le programme tpm2.

L'objectif est de définir et de visualiser les différentes aires de la surface d'affichage. A chacune d'elles sera associée une transformation de normalisation. Pour tracer un objet dans une telle aire on sélectionnera la transformation de normalisation adéquate avant d'exécuter les primitives de sortie graphique.

L'analyse de l'interface conduit à définir les transformations suivantes :

- tr 1 toute l'aire d'affichage
- tr 2 l'aire de tracé
- tr 3 partie supérieure de l'aire d'affichage
- tr 2 partie supérieure réduite
- tr 4 partie droite de l'aire d'affichage
- tr24 partie droite réduite
- tr 5 partie inférieure de l'aire d'affichage
- tr25 partie inférieure réduite
- tr 6 partie droite de l'aire d'affichage
- tr26 partie droite réduite



Les transformations de normalisation

On rédige à cet effet une routine appelée **marge** qui réalise uniquement la définition de ces zones. Quatre valeurs permettent de définir précisément les clôtures des transformations de normalisation associées. Ces quatre données correspondent à la largeur des marges autour de l'aire de tracé, soient **xmarg,xmard,ymarb,ymarh**. Ces variables sont définies dans un common étiqueté par le nom **defmar**.

common /defmar/xmarg,xmard,ymarb,ymarh

La routine **marge** n'aura pas de paramètre. Cette routine sera réutilisable dans bien des applications, seules les valeurs attribuées aux marges seront à définir.

De même, mais cette fois de manière propre à notre application, on définit les aires réservées à l'affichage des fonctions du logiciel. On rédige une routine nommée **dfct** qui va introduire pour chaque fonction une transformation de normalisation. On décide de programmer huit fonctions. Dans le programme principal on introduit une structure de données où seront décrites les clôtures associées à chaque transformation. Cinq valeurs suffisent à définir chaque transformation, soient le numéro de la transformation et les coordonnées des deux points extrêmes (inférieur gauche et supérieur droit). On affectera les numéros 31 à 38 à ces transformations.

call dfct(nbtr,xyfct)

nbtr : nb de transformations à traiter
xyfct : définition des nbtr transformations

Il ne reste plus qu'à visualiser sur la surface d'affichage toutes ces zones rectangulaires. Là encore on rédige un module réutilisable qui trace un rectangle dont les limites sont celles d'une transformation de normalisation donnée. Une fois ce module réalisé, il suffit de l'exécuter autant de fois que nécessaire pour matérialiser les diverses boîtes dont est constitué notre interface.

La routine de tracé sera appelée **visfen** et devra comporter les paramètres nécessaires à la définition des attributs du polygone à visualiser (un rectangle dans notre cas).

call visfen(ntr,.....)

ntr : numéro de transformation
.... :

Remarques:

Après chaque exécution du programme, il sera bon de vérifier le contenu du fichier des erreurs gkserr. Il ne doit contenir aucun message.

Sur PC, à supposer qu'un plotter HPGL y soit connecté, le même programme TPM2 est à même de réaliser cette fois sur le traceur de courbes le tracé de notre aire d'affichage. Là deux possibilités se présentent ; elles permettent de bien saisir l'utilité du fichier de sortie associé au poste de travail. Suivant le poste de travail choisi le tracé sera produit soit dans le fichier de sortie (spool), soit directement sur le périphérique.

Pour les plotter HPGL les numéros de poste de travail autorisés sous GKSbX sont :

- 7280 plotter HPGL (mode portrait), sortie dans un fichier
- 7281 plotter HPGL (mode paysage), sortie dans un fichier
- 717280 plotter HPGL (mode portrait), sortie directe sur com1

EXERCICE 3

On reprendra le texte du programme TPM2 pour produire TPM3. Dans cet exercice on va se consacrer à introduire le moniteur de tâches. Chaque tâche consistera dans un premier temps à afficher un message indiquant la nature de la fonction à exécuter. Ce moniteur sera piloté de préférence à l'aide de la souris.

Dans cette session de travaux pratiques on ne va définir que le contenu des boîtes suivantes :

- 31 Définition de l'axe X
- 32 Définition de l'axe Y
- 38 Fin de l'application

Cependant on prévoit dans notre programmation l'ensemble des huit fonctions. Pour inscrire un libellé représentatif de chaque fonction dans chaque boîte, il est conseillé de rédiger une routine adaptée à cet usage. Les libellés peuvent être limités à 10 caractères et seront contenus dans une structure de données du programme principal. La routine s'appellera t10fen et devra avoir les paramètres suivants :

call t10fen(ntr,libfct,.....)

ntr : numéro de la transformation
libfct : libellé de la fonction (10 caractères)
certains attributs de texte (police, précision, couleur)

Pour choisir une des fonctions de l'automate on utilisera la primitive d'entrée REQUEST LOCATOR.

Le schéma de ce moniteur est le suivant:

- 1) Initialisation du REQUEST LOCATOR
- 2) Requête au LOCATOR
- 3) Récupération d'un numéro de transformation (31 à 38)
- 4) Aiguillage suivant le numéro de transformation

Fct 1: afficher un message , attente et retour à 1
Fct 2: afficher un message , attente et retour à 1
Fct 3: afficher un message , attente et retour à 1
Fct 4: afficher un message , attente et retour à 1
Fct 5: afficher un message , attente et retour à 1
Fct 6: afficher un message , attente et retour à 1
Fct 7: afficher un message , attente et retour à 1
Fct 8: afficher un message , attente et suite

Le REQUEST LOCATOR est une des primitives d'entrée graphique de GKS. Comme chacune d'elle, il faut dans un premier temps l'initialiser. Cette initialisation consiste essentiellement à spécifier le numéro d'appareil utilisé (clavier, souris, ...), la position initiale du prompter associé à l'appareil choisi, la zone d'écho de ce prompter.

Sur PC, GKSbx propose deux appareils différents, chacun d'eux est référencé par un numéro :

1 clavier

touches utilisables
curseur haut (dép. verticaux)
curseur bas
curseur gauche (dép. horizontaux)
curseur droit
home (dép. diagonale)
fin
page précédente
page suivante
insertion (précision)
return (refus)
barre d'espacement (sélection)

2 souris

Sur SUN un seul appareil (numéro 1) est utilisable, c'est la souris

La zone d'écho est une zone rectangulaire de la surface d'affichage. Elle est donc définie par quatre nombres : le minimum, le maximum sur x et de même en y. La difficulté réside dans le fait que ces valeurs doivent être exprimées en coordonnées DC. La réponse à cette difficulté consiste à écrire une bonne fois pour toute une routine qui accepte en entrée les coordonnées en WC, le numéro de la transformation de normalisation associée et qui produit en sortie les valeurs converties en unités DC. Cette routine sera commentée en cours.

call zecho(nrm,xmi,xma,ymi,yma,x1,x2,y1,y2)

La position initiale du locator est par contre fournie en coordonnées universelles WC.

En retour d'une requête au locator on reçoit notamment un status qui rend compte de la validation (1) ou non (0) d'une position, et le numéro de transformation de normalisation dans laquelle la position a été validée. La position validée est renvoyée en coordonnées WC.

Le moniteur de tâches devra contrôler avec précision toutes ces valeurs afin que le programme soit robuste. Le numéro de transformation renvoyée servira à calculer un index qui pilotera l'aiguillage vers les diverses fonctions de l'application.

REMARQUES :

Après chaque exécution pendant la phase de mise au point, il est conseillé de vérifier le contenu du fichier gkserr.

Sur SUN la définition de la zone d'écho est plus floue dans la version actuelle de GKS_{Bx}. La gestion de la souris est plus complexe puisque partie intégrante de l'environnement SUNVIEW. On choisira comme zone d'écho toute la surface d'affichage. Cette restriction n'entrave en rien le bon fonctionnement de l'application pour peu qu'un test fin sur le numéro de transformation retournée soit judicieusement programmé.

EXERCICE 4

On reprendra le texte de TPM3 pour produire TPM4.

Dans cet exercice, on introduit une troisième fonction dans notre automate :

la fonction "tracer".

Le but du TP consiste à implémenter le contenu des trois tâches élémentaires :

définir un axe des abscisses, son libellé et les tracer

définir un axe des ordonnées et le tracer

tracer la courbe dans la fenêtre réservée à cet effet

Ces fonctions pourront être activées par l'opérateur dans n'importe quel ordre. Une contrainte apparaît immédiatement, le programme devra vérifier que les deux axes sont définis avant de tracer la courbe.

⇒ **La fonction 'axe X' :**

Pour implémenter cette fonction il faut introduire les valeurs extrêmes de l'axe ainsi que la valeur de l'intervalle séparant les graduations. Pour habiller cet axe on saisira un libellé de 50 caractères au maximum qui apparaîtra au dessous des valeurs associées à chaque graduation. Pour introduire ces diverses données on utilisera le REQUEST STRING. Cette primitive d'entrée est assez délicate à mettre en oeuvre, mais indispensable dans toute application interactive sous GKS. Quelques détails techniques sont fournis plus bas à propos de cet outil.

Le tracé de l'axe, de ses graduations et du titre ne pose pas de difficulté particulière. Des routines simples sont proposées dans le package BBGKS afin de gagner du temps de programmation. Il est conseillé d'étudier ces modules qui constituent des exemples de programmation des primitives de base telles que la polyligne, la gestion des attributs de texte, etc.

Une fois défini et tracé cet axe, on n'oubliera pas de positionner un indicateur qui informera la fonction 'tracé de courbe' de l'existence de l'axe X.

⇒ **La fonction 'axe Y' :**

On procédera de manière identique, mais on ne saisira pas de libellé.

⇒ **La fonction 'tracer' :**

A l'aide des valeurs extrêmes des deux axes il est possible de définir la fenêtre de la transformation de normalisation associée à l'aire de tracé, la clôture ayant été définie précédemment (tr. 2). Avant de réaliser cette opération il faudra s'assurer que les indicateurs associés aux axes sont à la bonne valeur. Le tracé proprement dit est sans intérêt, il suffit d'appeler le module trctf fourni dans BBGKS. La fonction est centrée en (0., 0.) et s'étend sur une aire limitée à quelques unités autour de l'origine. Avant de tracer cette fonction il sera bon de rafraîchir la zone de tracé.

Mise en œuvre du REQUEST STRING

Comme toute entrée graphique il faut procéder en deux temps :

- 1) initialisation
- 2) requête

Dans la phase d'initialisation on doit préciser l'appareil d'entrée (le clavier, 1), une chaîne initiale, la longueur du tampon d'entrée, la position de départ du curseur, le type d'écho retourné et la zone d'écho.

La détermination de la zone d'écho, exprimée en coordonnées DC, dépend de la longueur de la chaîne. On peut rédiger un module un peu plus général qui calcule les coordonnées DC d'une zone rectangulaire permettant l'affichage de n1 lignes de n2 caractères. Cette zone sera déterminée relativement à la clôture d'une transformation de normalisation donnée.

Dans l'exercice on introduira une nouvelle transformation (40) dont la clôture sera située dans la partie basse de la surface d'affichage, elle contiendra une ligne de texte. Cette zone de dialogue apparaîtra au besoin et disparaîtra après usage. Par hypothèse on considérera que toute la surface d'affichage (écran, fenêtre Sunview) peut contenir au plus 24 lignes de 80 caractères.

La chaîne initiale sera constituée de 50 caractères, les premiers invitant l'utilisateur à entrer les données désirées, par exemple pour l'axe X : 'xmin,xmax,xinc ?'. Les caractères restants seront identiques. On choisira judicieusement ce caractère afin de faciliter l'exploitation de la chaîne retournée par la requête. Le caractère '_' semble être, dans bien des cas, un bon caractère.

Lors de l'acquisition, la chaîne initiale est affichée, dans la limite du tampon, à l'intérieur de la zone d'écho. Le curseur est mis à la position de départ stipulée lors de l'initialisation et la phase de saisie démarre. Au retour, la chaîne acquise est constituée de la chaîne initiale modifiée par les caractères frappés sur l'organe d'entrée. Pour récupérer le message proprement dit, il faut donc l'extraire de la chaîne retournée. Afin de standardiser cette opération d'extraction, la rédaction d'une routine s'impose. Le kit BBGKS propose le module `extcha`, le caractère '_' y est considéré comme caractère de remplissage.

De même que pour le LOCATOR, le retour de la requête est qualifié par un status (la valeur 1 traduit un retour satisfaisant).

REMARQUES :

Comme pour les autres exercices, pendant la phase de mise au point, il faut veiller au contenu du fichier gkserr.

On remarquera que la disposition initiale des diverses fenêtres est à revoir légèrement pour se ménager un espace suffisant à la zone de dialogue 'texte'. Une simple modification de quelques valeurs dans les structures de données qui décrivent notre surface d'affichage permet immédiatement de redisposer les fenêtres ; il ne doit pas être nécessaire d'intervenir sur les instructions exécutables du programme si la programmation a été réalisée sainement.

EXERCICE 5

Ce dernier exercice sera consacré à l'implémentation des deux dernières fonctionnalités : l'effacement d'un axe au choix, la création d'un métatichier contenant le dessin final. Pour réaliser ce travail on reprendra le résultat de l'exercice précédent.

⇒ La fonction 'Efface Segment' :

Pour pouvoir sélectionner un objet dans le dessin déjà produit, et le détruire, l'utilisation du concept de segment associée à celle de la primitive d'entrée REQUEST PICK est une solution élégante et facile à mettre en œuvre. Le tracé d'un axe par exemple constituera un segment. Ces segments seront nommés séquentiellement à compter de 1.

Pour choisir l'axe à détruire, on utilisera la souris (ou le clavier). Le REQUEST PICK rapportera notamment un numéro de segment si on clique là où il faut. La destruction du segment entraînera sa disparition à l'écran. Il faudra veiller à repositionner à faux l'indicateur de définition d'axe. Pour cela il faut pouvoir déterminer si le segment désigné contient un axe des abscisses ou un axe des ordonnées. On peut utiliser une structure de donnée propre à l'application qui mémorise la table (numéro de segment, type d'axe), puis la consulter. GKS permet de résoudre la question plus simplement. On utilisera l'attribut 'identificateur de désignation' des primitives graphiques de sortie stockées dans le segment. Il suffit de choisir par exemple comme identificateur la valeur 7777 pour toutes les primitives de sortie qui produisent l'axe X et la valeur 8888 lors de la génération de l'axe Y.

Pour initialiser le relevage de désignation, comme pour les autres primitives d'entrée, on définira sa zone d'écho comme étant égale à toute la surface d'affichage (tr. 1). La position initiale du relevage sera désignée. Pour cela on prendra soin de créer un segment nommé 9999 contenant au moins une primitive de sortie graphique avec l'attribut 'indicateur de désignation' fixé à la valeur 9999 également. Ce segment ne servira qu'à l'initialisation du relevage et par conséquent sera détruit immédiatement au retour de la requête.

⇒ La fonction 'Création d'un métatichier'

Pour que cette fonction s'exécute correctement il faudra s'assurer que les axes sont correctement définis. On aura pris soin de conserver les deux segments correspondants dans le WISS (Worksation Independent Storage Segment) ainsi que leur nom lors de leur création. L'entrée du nom de fichier associé sera faite par une requête au relevage de chaîne, on utilisera la même zone de dialogue que pour la définition des axes.

Pour créer ce métatichier, on devra ouvrir un nouveau poste de travail de type MO. On recopiera les deux segments contenant les axes sur ce poste, puis on procédera au tracé de la courbe dans sa fenêtre. On désactivera pendant cette période le poste de travail associé à la console. Le poste de travail de type MO sera enfin désactivé puis fermé.

Le fichier résultant pourra être testé en utilisant l'utilitaire LECMET. Le code source de ce programme sera donné aux participants. Il sera bon de visualiser une petite partie du contenu de ce métichier afin de mémoriser sa structure. La lecture et la compréhension du code de LECMET en seront facilitées.

FONCTIONS GKS**FONCTIONS DE CONTROLE**

GOPKS (errfil,buf)	open gks
GCLKS	close gks
GOPWK (wkid,conid,wtype)	open workstation
GCLWK (wkid)	close workstation
GACWK (wkid)	activate workstation
GDAWK (wkid)	deactivate workstation
GCLRWK (wkid,coff)	clear workstation
GRSGWK (wkid)	redraw all segments on workstation
GUWK (wkid,regfl)	update workstation
GSDS (wkid,defrmod,regmod)	set deferral state
GMSG (wkid,mess)	message
GESC (fctid,lidr,datrec)	escape

FONCTIONS DE SORTIE

GPL (n,px,py)	polyline
GPM (n,px,py)	polymarker
GTX (px,py,chars)	text
GFA (n,px,py)	fill area
GCA (px,py,qx,qy,dx,dy,isc,isr,ddx,ddy,colia)	cell array
GGDP (n,px,py,primid,lidr,datrec)	generalized drawing primitive

ATTRIBUTS DE SORTIE

ATTRIBUTS DE PRIMITIVES INDEPENDANT DU POSTE DE TRAVAIL

GSPLI (index)	set polyline index
GSLN (ltype)	set linetype
GSLWSC (width)	set linewidth scale factor
GSPLCI (col)	set polyline color index
GSPMI (index)	set polymarker index
GSMK (mtype)	set markertype
GSMKSC (mszsf)	set marker size scale factor
GSPMCI (col)	set polymarker color index
GSTXI (index)	set text index
GSTXFP (font,prec)	set text font and precision
GSCHXP (chxp)	set character expansion factor
GSCHSP (chsp)	set character spacing
GSTXCI (col)	set text color index
GSCHIH (chh)	set character height
GSCHUP (chux,chuy)	set character up vector
GSTXP (txp)	set text path
GSTXAL (txalh,txalv)	set text alignment
GSFAI (index)	set fill area index
GSFAIS (ints)	set fill area interior style
GSFASI (styl)	set fill area style index
GSFACI (col)	set fill area color index
GSPA (szx,szy)	set pattern size
GSPARF (rfx,rfy)	set pattern reference point
GSASF (last)	set aspect source flags
GSPKID (pcid)	set pick identifier

ATTRIBUTS DE PRIMITIVES DEPENDANT DU POSTE DE TRAVAIL

GSPLR (wkid,pli,ltype,lwidth,coli)	set polyline representation
GSPMR (wkid,pmi,mtype,mszsf,coli)	set polymarker representation
GSTXR (wkid,tci,font,prec,chxp,chsp,coli)	set text representation
GSFAR (wkid,fai,ints,styl,coli)	set fill area representation
GSPAR (wkid,pai,dx,dy,dimx,isc,isr,dx,dy,colia)	set pattern representation
GSCR (wkid,ci,cr,og,cb)	set color representation

FONCTIONS DE TRANSFORMATION

TRANSFORMATION DE NORMALISATION WC - NDC

GSWN (tnr,xmin,xmax,ymin,ymax)	set window
GSVP (tnr,xmin,xmax,ymin,ymax)	set viewport
GSVPIP (tnr,rtnr,relpri)	set viewport input priority
GSELNT (tnr)	select normalization transformation
GSCLIP (clsw)	set clipping indicator

TRANSFORMATION DE POSTE DE TRAVAIL NDC - DC

GSWKWN (wkid,xmin,xmax,ymin,ymax)	set workstation window
GSKWVP (wkid,xmin,xmax,ymin,ymax)	set workstation viewport

FONCTIONS DE SEGMENTATION

FONCTIONS DE MANIPULATION DE SEGMENT

GCRSG (sgna)	create segment
GCLSG	close segment
GRENSG (old,new)	rename segment
GDSG (sgna)	delete segment
GDSGWK (wkid,sgna)	delete segment from workstation
GASGWK (wkid,sgna)	associate segment with workstation
GCSGWK (wkid,sgna)	copy segment to workstation
GINSG (sgna,m)	insert segment

ATTRIBUTS DE SEGMENT

GSSGT (sgna,m)	set segment transformation
GSVIS (sgna,vis)	set visibility
GSHLT (sgna,hil)	set highlighting
GSSGP (sgna,prior)	set segment priority
GSDTEC (sgna,det)	set detectability

FONCTIONS D'ENTREE

FONCTIONS D'INITIALISATION DES APPAREILS D'ENTREE

GINLC(wkid,lcnbr,tnr,ipx,ipy,pet,xmin,xmax,ymin,ymax,ldr,datrec)	initialize locator
GINSK(wkid,skdnr,tnr,n,ipx,ipy,pet,xmin,xmax,ymin,ymax,buflen,ldr,datrec)	initialize stroke
GINVL(wkid,vldnr,ival,pet,xmin,xmax,ymin,ymax,loval,hival,ldr,datrec)	initialize valuator
GINCH(wkid,chdnr,istat,ichnr,pet,xmin,xmax,ymin,ymax,ldr,datrec)	initialize choice
GINPK(wkid,pcdnr,istat,sgna,ipcid,pet,xmin,xmax,ymin,ymax,ldr,datrec)	initialize pick
GINST(wkid,stdnr,istr,pet,xmin,xmax,ymin,ymax,buflen,inipos,ldr,datrec)	initialize string

FONCTIONS DE REQUETE

GRQLC (wkid,lcnbr, stat,tnr,px,py)	request locator
GRQSK (wkid,skdnr,n, stat,tnr,np,px,py)	request stroke
GRQVL (wkid,vldnr, stat, val)	request valuator
GRQCH (wkid,chdnr, stat,chnr)	request choice
GRQPK (wkid,pcdnr, stat,sgna,pcid)	request pick
GRQST (wkid,stdnr, stat,lostr,str)	request string

MODE DES APPAREILS D'ENTREE

GSLCM (wkid,lcnbr,mode,esw)	set locator mode
GSSKM (wkid,skdnr,mode,esw)	set stroke mode
GSVLM (wkid,vldnr,mode,esw)	set valuator mode
GSCHM (wkid,chdnr,mode,esw)	set choice mode
GSPKM (wkid,pcdnr,mode,esw)	set pick mode
GSSTM (wkid,stdnr,mode,esw)	set string mode

FONCTIONS DE METAFICHIER

GWITM (wkid,type,idrl,ldr,datrec)	write item to gksm
GGTTM (wkid,type,ldr)	get item type from gksm
GRDTM (wkid,mldr,ldr,datrec)	read item from gksm
GIITM (type,idrl,ldr,datrec)	interpret item

FONCTIONS D'INTERROGATION

FONCTION D'INTERROGATION DU MODE OPERATOIRE

GQOPS (opsta)	inquire operating state value
----------------------	-------------------------------

FONCTIONS D'INTERROGATION DES POSSIBILITES DE L'IMPLEMENTATION GKS_{Bx}

GQLVKS (errind,level)	inquire level of gks
GQEWK (n, errind,number,ktyp) workstation types	inquire list element of available
GQWKM (errind,mxopwk,mcacwk,mxwkas)	inquire workstation maximum number
GQMNTN (errind,maxtnr)	inquire maximum normalization number

FONCTIONS D'INTERROGATION DE LA LISTE D'ETAT DE GKS

GQOPWK (n, errind,ol,wkid)	inquire set member of open workstations
GQACWK (n, errind,ol,wkid)	inquire set member of active workstations
GQPKID (errind,pkid)	inquire pick identifier
GQPLI (errind,index)	inquire polyline index
GQPMI (errind,index)	inquire polymarker index
GQTXI (errind,index)	inquire text index
GQCHH (errind,chh)	inquire character height
GQCHUP (errind,chux,chuy)	inquire character up vector
GQCHB (errind,chbx,chby)	inquire character base vector
GQTX (errind,txp)	inquire text path
GQTXAL (errind,txalh,txalv)	inquire text alignment
GQFAI (errind,index)	inquire fill area index
GQLN (errind,ltype)	inquire linetype
GQLWSC (errind,width)	inquire linewidth scale factor

GQPLCI (errind,col)	inquire polyline color index
GQMKT (errind,mtype)	inquire markertype
GQMKS (errind,mszsf)	inquire marker size scale factor
GQPMCI (errind,col)	inquire polymarker color index
GQTXF (errind,font,prec)	inquire text font and precision
GQCHXP (errind,chxp)	inquire character expansion factor
GQCHW (errind,chw)	inquire character width
GQCHSP (errind,chsp)	inquire character spacing
GQTXCI (errind,col)	inquire text color index
GQFAIS (errind,ints)	inquire fill area interior style
GQFASI (errind,styl)	inquire fill area style index
GQFACI (errind,col)	inquire fill area color index
GQPA (errind,pwx,pwy,phx,phy)	inquire pattern size
GQPARF (errind,fx,fy)	inquire pattern reference point
GQASF (errind,lasf)	inquire aspect source flags
GQCNTN (errind,ctnr)	inquire current normalization transformation number
GQENTN (n, errind,ol,fpri)	inquire list element of normalization transformation number
GQNT (ntr, errind,window,viewp)	inquire normalization transformation
GQCLIP (errind,clip)	inquire clipping indicator
GQOPSG (errind,segnam)	inquire name of open segment
GQSGUS (n, errind,ol,segnam)	inquire set member of segment

**FONCTIONS D'INTERROGATION DE LA LISTE D'ETAT
D'UN POSTE DE TRAVAIL**

GQWKC (wkid, errind,conid,wtype)	inquire workstation connection and type
GQWKS (wkid, errind,state)	inquire workstation state
GQWKDU (wkid, errind,defmod,regmod,deempty,nframe)	inquire workstation deferral and update states
GQEPLI (wkid,n, errind,ol,plind)	inquire list element of polyline indices
GQPLR (wkid,pli,type, errind,intype,lwidth,coli)	inquire polyline representation
GQEPMI (wkid,n, errind,ol,pmind)	inquire list element of polymarker indices
GQPMR (wkid,pmi,type, errind,mktype,mksscf,coli)	inquire polymarker representation
GQETXI (wkid,n, errind,ol,txind)	inquire list element of text index
GQTXR (wkid,txi,type, errind,font,prec,charxp,charsp,coli)	inquire text representation
GQTXX (wkid,px,py,str, errind,cpx,cpy,txexp,txexp)	inquire text extent
GQEFAI (wkid,n, errind,ol,faind)	inquire list element of fill area indices
GQFAR (wkid,fai,type, errind,style,stylid,coli)	inquire fill area representation
GQEPAI (wkid,n, errind,ol,paind)	inquire list element of pattern
GQPAR (wkid,pai,type,nmx,mmx,errind,n,m,parray)	inquire pattern representation

GQECl (wkid,n, errind,ol,colind)	inquire list element of color index
GQCR (wkid,coli,type, errind,red,green,blue)	inquire color representation
GQWKT (wkid, errind,tus,rwindo,cwindo,rviewp,cviewp)	inquire workstation transformation
GQSGWK (wkid,n, errind,ol,segnam)	inquire set member of segment names on workstation
GQLCS (wkid,lcdnr,type,mldr, errind,mode,esw, itnr,tlpx,tlpv,pet,earea,ldr,datrec)	inquire locator device state
GQSKS (wkid,skdnr,type,n,mldr,errind,mode, esw,itnr,np,px,py,pet,earea,buflen,ldr,datrec)	inquire stroke device state
GQVLS (wkid,vldnr,mldnr, errind,mode, esw,ival,pet,earea,loval,lval,ldr,datrec)	inquire valuator device state
GQCHS (wkid,chdnr,mldr, errind,mode, esw,ichnr,pet,earea,ldr,datrec)	inquire choice device state
GQPKS (wkid,pcdnr,type,mldr, errind,mode, esw,istat,isgna,ipcid,pet,earea,ldr,datrec)	inquire pick device state
GQSTS (wkid,stdnr,mldr, errind,mode,esw, losr,istr,pet,earea,buflen,inipos,ldr,datrec)	inquire string device state

FONCTIONS D'INTERROGATION DES TABLES DE DESCRIPTION DE POSTE DE TRAVAIL

GQWKCA (wtype, errind,wkcat)	inquire workstation category
GQWKCK (wtype, errind,vrtype)	inquire workstation classification
GQDSP (wtype, errind,dcunit,rx,ry,bx,ly)	inquire maximum display surface size
GQDWKA (wtype, errind,plbun,pmbun, txbun,fabun,parep,colrep,wktr)	inquire dynamic modification of workstation attributes
GQDDDS (wtype, errind,defmod,regmod)	inquire default deferral state
GQPLF (wtype,n, errind,nit,lt,rlw,nomlw, rlwmin,rlwmax,nppli)	inquire polyline facilities
GQPPLR (wtype,pli, errind,intype,lwidth,coli)	inquire predefined polyline representation
GQPMF (wtype,n, errind,nmt,mt,nms,nomms, rmsmin,rmsmax,npymi)	inquire polymarker facilities
GQPPMR (wtype,pmi, errind,mkttype,mksscf,coli)	inquire predefined polymarker representation
GQTXF (wtype,n,rrind,rfpp,font,prec,nchh, minchh,maxchh,nchx,minchx,maxchx,cptxi)	inquire text facilities
GQPTXR (wtype,ptxi, errind,font,prec, charxp,charsp,coli)	inquire predefined text representation
GQFAF (wtype,ni,nh, errind,nis,is,nhs,hs,npfai)	inquire fill area facilities
GQPFAR (wtype,pfai, errind,style,stylid,coli)	inquire predefined fill arearepresentation

GQPAF (wtype, errind,nppai)	inquire pattern facilities
GQPPAR (wtype,ppai,nmx, errind,n,m,parray)	Inquire predefined pattern representation
GQCF (wtype, errind,ncoli,cola,npcl)	inquire color facilities
GQPCR (wtype,pci, errind,red,green,blue)	inquire predefined color representation
GQE GDP (wtype,n, errind,ngdp,gdp)	inquire list element of available generalized primitives
GQGDP (wtype,gdp, errind,nbnd,bndl)	inquire generalized drawing primitive
GQLWK (wtype, errind,mplbte,mpmbte, mtxbte,mtfalte,mpai,mcoli)	Inquire maximum length of workstation state tables
GQS GP (wtype, errind,ngs)	inquire number of segment priorities supported
GQDSGA (wtype, errind,sgtr,vonoff, voffon,high,sgpr,add,sgdel)	inquire dynamic modification of segment attributes
GQLI (wtype, errind,nlcd,nskd,nvid, nchd,npcd,nstd)	inquire number of available logical input devices
GQDLC (wtype,devnd,n,midr, errind,dpx,dpy, oi,pet,earea,ldr,datrec)	inquire default locator device data
GQDSK (wtype,devno,n,midr, errind,dbufsk, oi,pet,earea,buflen,ldr,datrec)	inquire default stroke device data
GQDVL (wtype,devno,n,midr, errind,dval, oi,pet,earea,loval,hival,ldr,datrec)	inquire default valuator device data
GQDCH (wtype,devno,n,midr, errind, malt,oi,petl,earea,ldr,datrec)	inquire default choice device data
GQDPK (wtype,devno,n,midr, errind, oi,petl,earea,ldr,datrec)	Inquire default pick device data
GQDST (wtype,devno,n,midr, errind,mruff, oi,pet,area,buflen,inipos,ldr,datrec)	inquire default string device data

FONCTIONS D'INTERROGATION DE LA LISTE D'ETAT DES SEGMENTS

GQASWK (sgna,n, errind,oi,wkid)	inquire set member of associated workstations
GQSGA (sgna, errind,segtm,vis,high,sgpr,det)	inquire segment attributes

FONCTIONS D'INTERROGATION PIXEL

GQPXAD (wkid,px,py,qx,qy, errind,n,m)	inquire pixel array dimensions
GQPXA (wkid,px,py,dx,dy,dimx,errind,inval,colia)	inquire pixel array
GQPX (wkid,px,py, errind,coli)	inquire pixel

UTILITAIRES

GEVTM (xo,yo,dx,dy,phi,fx,fy,sw,mout)	evaluate transformation matrix
GACTM (min,xo,yo,dx,dy,phi,fx,fy,sw,mout)	accumulate transformation matrix
GPREC (il,ia,rl,ra,sl,lstr,str,midr,errind,lidr,datrec)	pack data record
GUREC (lidr,datrec,iil,irl,isl,errind,il,ia,rl,ra,sl,lstr,str)	unpack data record

FONCTIONS DE GESTION DES ERREURS

GECLKS	emergency close gks
GERHND (errnr,fctid,errfil)	error handling
GERLOG (errnr,fctid,errfil)	error logging

LISTE DES ERREURS GKS_{Bx}**ETATS**

- 1 gks not in proper state: gks shall be in the state gkcl
- 2 gks not in proper state: gks shall be in the state gkop
- 3 gks not in proper state: gks shall be in the state wsac
- 4 gks not in proper state: gks shall be in the state sgop
- 5 gks not in proper state: gks shall be either in the state wsac or in the state sgop
- 6 gks not in proper state: gks shall be either in the state wsop or in the state wsac
- 7 gks not in proper state: gks shall be in one of the states wsop,wsac or sgop
- 8 gks not in proper state: gks shall be in one of the states gkop,wsoc wsac or sgop

STATIONS DE TRAVAIL

- 20 specified workstation identifier is invalid
- 21 specified connection identifier is invalid
- 22 specified workstation type is invalid
- 23 specified workstation type does not exist
- 24 specified workstation is open
- 25 specified workstation is not open
- 26 specified workstation cannot be opened
- 27 workstation independent segment storage is not open
- 28 workstation independent segment storage is already open
- 29 specified workstation is active
- 30 specified workstation is not active
- 31 specified workstation is of category mo
- 32 specified workstation is not of category mo
- 33 specified workstation is of category mi
- 34 specified workstation is not fo category mi
- 35 specified workstation is of category input
- 36 specified workstation is workstation independent segment storage
- 37 specified workstation is not of category outin
- 38 specified workstation is neither of category input nor of category output
- 39 specified workstation is neither of category output nor of category outin
- 40 specified workstation has no pixel store readback capability
- 41 specified workstation type is not able to generate the specified generalized drawing primitive.
- 42 maximum number of simultaneously open workstations would be exceeded
- 43 maximum number of simultaneously active workstations would be exceeded

TRANSFORMATIONS

50 transformation number is invalid
51 rectangle definition is invalid
52 viewport is not within the normalized device coordinate unit square
53 workstation window is not within the normalized device coordinate unit square
54 workstation viewport is not within the display space

ATTRIBUTS DE SORTIE

60 polyline index is invalid
61 a representation for the specified polyline index has not been define on this workstation
63 linetype is less than or equal to zero
64 specified linetype is not supported on this workstation
65 line width scale factor is less than zero
66 polymarker index is invalid
69 marker type is less than or equal to zero
70 specified marker type is not supported on this workstation
71 marker size scale factor is less than zero
72 text index is invalid
73 up vector is equal to zero
75 text font is less than or equal to zero
76 requested text font is not supported for the specified precision on workstation
77 character expansion factor is less than or equal to zero
78 character height is less than or equal to zero
79 length of character up vector is zero
80 fill area index is invalid
83 specified fill area interior style is not supported on this workstation
84 style (pattern or hatch) index is less than or equal to zero
85 specified pattern index is invalid
86 specified hatch style is not supported on this workstation
90 interior style pattern is not supported on this workstation
91 dimensions of color array are invalid
92 color index is less than zero
93 color index is invalid
96 color is outside range (0,1)
97 pick identifier is invalid

PRIMITIVES DE SORTIE

- 100 number of points is invalid
- 101 invalid code in string
- 102 generalized drawing primitive identifier is invalid
- 103 content of generalized drawing primitive data record is invalid
- 104 at least one active workstation is not able to generate the specified generalized drawing primitive

SEGMENTS

- 120 specified segment name is invalid
- 121 specified segment name is already in use
- 122 specified segment does not exist
- 123 specified segment does not exist on specified workstation
- 124 specified segment does not exist on workstation independent segment storage
- 125 specified segment is open
- 126 segment priority is outside the range (0,1)

APPAREILS D'ENTREE

- 140 specified input device is not present on workstation
- 141 input device is not in request mode
- 142 input device is not in sample mode
- 143 event and sample input mode are not available at this level of gks
- 144 specified prompt and echo type is not supported on this workstation
- 145 echo area is outside display space
- 146 contents of input data record are invalid
- 147 input queue has overflowed
- 148 input queue has not overflowed since gks was opened or the last invocation of inquire input queue overflow
- 149 input queue has overflowed, but associated workstation has been
- 150 no input value of the correct class is in the current event report
- 151 timeout is invalid
- 152 initial value is invalid
- 153 length of initial string is greater than the implementation define maximum

METAFICHIER

- 160 item type is not allowed for user items
- 161 item length is invalid
- 162 no item is left in gks metafile input
- 163 metafile item is invalid
- 164 item type is not a valid gks item
- 165 content of item data record is invalid for the specified item type
- 166 maximum item data record length is invalid
- 167 user item cannot be interpreted
- 200 specified error file is invalid

FONCTIONS D'ECHAPPEMENT

- 180 specified function is not supported
- 181 contents of escape data record are invalid
- 182 contents of escape data record are invalid

DEPENDANT DE L'IMPLEMENTATION

- 300 specified function is not supported in this level of gks
- 301 storage overflow has occurred in gks
- 302 storage overflow has occurred in segment storage
- 307 input/output error has occurred during program library management
- 308 input/output error has occurred while receiving data from a workstation
- 312 input/output error has occurred while sending data to a workstation
- 314 specified font not accessed, execution continuing, font 1 assumed
- 381 gks memory management error : overflow
- 382 gks memory management error : access memory not allowed
- 383 gks memory management error: read access memory not allowed
- 384 gks memory management error : writing capacity exceeded

LES TABLES DE GKS

Etat opératoire.

Table de description de GKS.

Elle contient la description des possibilités de la réalisation.

Table d'état de GKS.

Elle contient les variables d'état de GKS qui déterminent l'effet d'une procédure GKS.

Table d'état du poste de travail.

Chaque poste de travail ouvert possède une liste d'état. Pour les postes de travail de catégorie 'poste de travail META FICHIER EN SORTIE' certaines valeurs dépendent, en réalité, de la réalisation.

Table de description de poste de travail.

Les tables de description de poste de travail ne peuvent pas être modifiées par le programme d'application. Il n'existe qu'une table par type de poste disponible dans une installation donnée.

Table d'état des segments.

Il existe une table d'état des segments pour le segment ouvert et pour chacun des segments stockés.

Table d'état des erreurs.

Une description détaillée de ces tables est donnée dans :

Infographie Système graphique de base (GKS)
Description fonctionnelle
NF EN 27942 Décembre 1986 pp. 166-174

QUELQUES ROUTINES DU PACKAGE BBGKS

Les sources des routines suivantes seront distribuées aux participants pendant les sessions de TP.

**marges
opgks
opwk (versions PC, SUN)
t10fen
texfen
vistfen**

⇨ Les COMMON de BBGKS

```
c
c      include des common de BBGKS.LIB
c
common / essos /    ios,namf,codop
common / systm /   rx,ry,rtx,rty
common /defmar/ xmarg,xmard,ymarb,ymarh
common /atxtxb/ high5,expa5,espa5,ialx5,ialy5,chupx5,
1          chupy5,ipath5,ipol5,iprec5,icol5
common /atxtxh/ high3,expa3,espa3,ialx3,ialy3,chupx3,
1          chupy3,ipath3,ipol3,iprec3,icol3
common /atxtyg/ high6,expa6,espa6,ialx6,ialy6,chupx6,
1          chupy6,ipath6,ipol6,iprec6,icol6
common /atxtyd/ high4,expa4,espa4,ialx4,ialy4,chupx4,
1          chupy4,ipath4,ipol4,iprec4,icol4
c
```

⇨ Tracé des axes

```
subroutine axexb(ntr,vxmin,vxmax,vxinc,yltic)
c
c -----
c
c          Bibliothèque BBGKS
c
c -----
c Atelier Info. Brest JJ Lechauve 1989
c
c But:      Trace d'un axe horizontal bas
c -----
c      ntr      E I4 : Numéro de la fenêtre
c      vxmin    E R4 : Val. min sur l'axe
c      vxmax    E R4 : Val. max sur l'axe
c      vxinc    E R4 : Val. du pas ( tic marque )
c      yltic    E R4 : Largeur de la tic marque
c
c      real x(2),y(2),win(4),view(4)
c      call gqnt(ntr,ier,win,view)
c      x(1)=win(1)
c      x(2)=win(2)
c      y(1)=win(4)
c      y(2)=win(4)
c      call gqcnty(ierr,ntra)
c      call gselnt(ntr)
c      call gpl(2,x,y)
c      dvx=vxmax-vxmin
c      dwx=win(2)-win(1)
c      dd=dwx/dvx*vxinc
c      x(1)=win(1)
c      y(1)=win(4)
c      xx=x(1)
c      call ticb(1,x,y,yltic)
10     xx=xx+dd
c      if(xx.gt.win(2)) goto 20
c      x(1)=xx
c      call ticb(1,x,y,yltic)
c      goto 10
20     call gselnt(ntra)
      return
      end
```

```
subroutine axexh(ntr,vxmin,vxmax,vxinc,yltic)
c
c -----
c
c          Bibliothèque BBGKS
c
c -----
c          Atelier Info. Brest JJ Lechauve 1989
c
c          But:      Trace d'un axe horizontal haut
c -----
c          ntr       E I4 : Numéro de la fenêtre
c          vxmin    E R4 : Val. min sur l'axe
c          vxmax    E R4 : Val. max sur l'axe
c          vxinc    E R4 : Val. du pas ( tic marque )
c          yltic    E R4 : Largeur de la tic marque
c
c          real x(2),y(2),win(4),view(4)
c          call gqnt(ntr,ier,win,view)
c          x(1)=win(1)
c          x(2)=win(2)
c          y(1)=win(3)
c          y(2)=win(3)
c          call gqcntn(ierr,ntra)
c          call gselnt(ntr)
c          call gpl(2,x,y)
c          dvx=vxmax-vxmin
c          dwx=win(2)-win(1)
c          dd=dwx/dvx*vxinc
c          x(1)=win(1)
c          y(1)=win(3)
c          xx=x(1)
c          call tich(1,x,y,yltic)
10       xx=xx+dd
c          if(xx.gt.win(2)) goto 20
c          x(1)=xx
c          call tich(1,x,y,yltic)
c          goto 10
20       call gselnt(ntra)
c          return
c          end
```

```
subroutine axeyd(ntr,vymin,vymax,vyinc,xltic)
c
c -----
c
c          Bibliothèque BBGKS
c
c -----
c          Atelier Info. Brest JJ Lechauve 1989
c
c          But:      Trace d'un axe vertical à droite
c -----
c          ntr       E I4 : Numéro de la fenêtre
c          vymin    E R4 : Val. min sur l'axe
c          vymax    E R4 : Val. max sur l'axe
c          vyinc   E R4 : Val. du pas ( tic marque )
c          xltic   E R4 : Largeur de la tic marque
c
c          real x(2),y(2),win(4),view(4)
c          call gqnt(ntr,ier,win,view)
c          x(1)=win(1)
c          x(2)=win(1)
c          y(1)=win(3)
c          y(2)=win(4)
c          call gqcntn(ierr,ntra)
c          call gselnt(ntr)
c          call gpl(2,x,y)
c          dvx=vymax-vymin
c          dwy=win(4)-win(3)
c          dd=dwy/dvx*vyinc
c          x(1)=win(1)
c          y(1)=win(3)
c          yy=y(1)
c          call ticd(1,x,y,xltic)
10         yy=yy+dd
c          if(yy.gt.win(4)) goto 20
c          y(1)=yy
c          call ticd(1,x,y,xltic)
c          goto 10
20         call gselnt(ntra)
c          return
c          end
```

```
subroutine axeyg(ntr,vymin,vymax,vyinc,xltic)
c
c -----
c
c          Bibliothèque BBGKS
c
c -----
c          Atelier Info. Brest JJ Lechauve 1989
c
c          But:      Trace d'un axe vertical a gauche
c -----
c          ntr       E I4 : Numéro de la fenêtre
c          vymin    E R4 : Val. min sur l'axe
c          vymax    E R4 : Val. max sur l'axe
c          vyinc   E R4 : Val. du pas ( tic marque )
c          xltic   E R4 : Largeur de la tic marque
c
c          real x(2),y(2),win(4),view(4)
c          call gqnt(ntr,ier,win,view)
c          x(1)=win(2)
c          x(2)=win(2)
c          y(1)=win(3)
c          y(2)=win(4)
c          call gqcntn(ierr,ntra)
c          call gselnt(ntr)
c          call gpl(2,x,y)
c          dvx=vymax-vymin
c          dwy=win(4)-win(3)
c          dd=dwy/dvx*vyinc
c          x(1)=win(2)
c          y(1)=win(3)
c          yy=y(1)
c          call ticg(1,x,y,xltic)
10       yy=yy+dd
c          if(yy.gt.win(4)) goto 20
c          y(1)=yy
c          call ticg(1,x,y,xltic)
c          goto 10
20       call gselnt(ntra)
c          return
c          end
```

⇨ Tracé des tic-marques sur les axes

```
subroutine ticb(n,x,y,ylgth)
c
c -----
c
c           Bibliothèque BBGKS
c
c -----
c Atelier Info. Brest JJ Lechauve 1989
c
c But:      Trace d'une tic-marque vers le bas
c -----
c     n          E I4 : Nombre de points (tic-marques)
c     x          E R4 : Abscisses sur l'axes
c     y          E R4 : Ordonnées
c     ylgth     E R4 : Taille de la tic-marque
c
c dimension x(1),y(1),xx(2),yy(2)
c do 10 i=1,n
c   xx(1)=x(i)
c   xx(2)=x(i)
c   yy(1)=y(i)
c   yy(2)=y(i)-ylgth
c   call gpl(2,xx,yy)
10  continue
c   return
c end

subroutine ticd(n,x,y,xlgth)
c
c -----
c
c           Bibliothèque BBGKS
c
c -----
c Atelier Info. Brest JJ Lechauve 1989
c
c But:      Trace d'une tic-marque vers la droite
c -----
c     n          E I4 : Nombre de points (tic-marques)
c     x          E R4 : Abscisses sur l'axes
c     y          E R4 : Ordonnées
c     xlgth    E R4 : Taille de la tic-marque
c
c dimension x(1),y(1),xx(2),yy(2)
c do 10 i=1,n
c   xx(1)=x(i)
c   xx(2)=x(i)+xlgth
c   yy(1)=y(i)
c   yy(2)=y(i)
c   call gpl(2,xx,yy)
10  continue
c   return
c end
```

```

subroutine ticg(n,x,y,xlgth)
c
c -----
c
c           Bibliothèque BBGKS
c
c -----
c   Atelier Info. Brest JJ Lechauve 1989
c
c   But:      Trace d'une tic-marque vers la gauche
c   ----
c       n          E I4 : Nombre de points (tic-marques)
c       x          E R4 : Abscisses sur l'axes
c       y          E R4 : Ordonnees
c       xlgth     E R4 : Taille de la tic-marque
c
c dimension x(1),y(1),xx(2),yy(2)
do 10 i=1,n
  xx(1)=x(i)
  xx(2)=x(i)-xlgth
  yy(1)=y(i)
  yy(2)=y(i)
  call gpl(2,xx,yy)
10 continue
return
end

```

```

subroutine tich(n,x,y,ylgth)
c
c -----
c
c           Bibliothèque BBGKS
c
c -----
c   Atelier Info. Brest JJ Lechauve 1989
c
c   But:      Trace d'une tic-marque vers le haut
c   ----
c       n          E I4 : Nombre de points (tic-marques)
c       x          E R4 : Abscisses sur l'axes
c       y          E R4 : Ordonnees
c       ylgth     E R4 : Taille de la tic-marque
c
c dimension x(1),y(1),xx(2),yy(2)
do 10 i=1,n
  xx(1)=x(i)
  xx(2)=x(i)
  yy(1)=y(i)
  yy(2)=y(i)+ylgth
  call gpl(2,xx,yy)
10 continue
return
end

```

⇨ Tracé des valeurs sur les axes

```

subroutine valxb(ntr,vxmin,vxmax,vxinc,ninc,fmt,ytx)
c
c -----
c          Bibliothèque BBGKS
c
c -----
c Atelier Info. Brest JJ Lechauve 1989
c
c But:      Ecriture des valeurs sur axe horizontal bas
c -----
c          ntr      E I4 : Numéro de la fenêtre
c          vxmin   E R4 : Val. min.
c          vxmax   E R4 : Val. max.
c          vxinc   S R4 : Val. increment
c          ninc    S I4 : Fréquence d inscription
c          fmt     E C15: Format de codage des nombres
c          ytx     E R4 : Ordonnée (NDC) pied des car.
c
character * 15 strg,fmt
common / systm / rx,ry,rtx,rty
common / defmar / xmarg,xmard,ymarb,ymarh
common / atxtxb/ high5,expa5,espa5,ialx5,ialy5,chupx5,
1           chupy5,ipath5,ipol5,iprec5,icols5
if (vxmax.lt.vxmin) goto 15
if (xmard.lt.xmarg) goto 15
x0=xmarg
x1=xmard
dx=x1-x0
dvx=vxmax-vxmin
xinc=dx/dvx*vxinc
c
call patrtx(high5,expa5,espa5,ialx5,ialy5,chupx5,chupy5,
&                   ipath5,ipol5,iprec5,icols5)
c
call gqcntn(ierr,ntra)
call gselnt(ntr)
y=ytx
i=0
3  v=vxmin+float(i)*vxinc
  if (v.gt.vxmax) goto 10
  x=x0+float(i)*xinc
  call cvfmt(v,fmt,strg,1)
  if (l.ne.0) call gtx(x,y,strg(1:1))
8  i=i+1
  if (mod(i,ninc).ne.0) goto 8
  goto 3
10 call ratrtx
  call gselnt(ntra)
15 return
end

```

```

subroutine valxh(ntr,vxmin,vxmax,vxinc,ninc,fmt,ytx)
c
c -----
c
c           Bibliothèque BBGKS
c
c -----
c       Atelier Info. Brest JJ Lechauve 1989
c
c       But:      Ecriture des valeurs sur axe horizontal haut
c -----
c           ntr      E I4 : Numéro de la fenêtre
c           vxmin   E R4 : Val. min.
c           vxmax   E R4 : Val. max.
c           vxinc   S R4 : Val. increment
c           ninc    S I4 : Fréquence d inscription
c           fmt     E C15: Format de codage des nombres
c           ytx     E R4 : Ordonnée (NDC) pied des car.
c
character * 15 strg,fmt
common / systm / rx,ry,rtx,rty
common / defmar / xmarg,xmard,ymarb,ymarh
common / atxtxh/ high3,expa3,espa3,ialx3,ialy3,chupx3,
1          chupy3,ipath3,ipol3,iprec3,icol3
if (vxmax.lt.vxmin) goto 15
if (xmard.lt.xmarg) goto 15
x0=xmarg
x1=xmard
dx=x1-x0
dvx=vxmax-vxmin
xinc=dx/dvx*vxinc
c
call patrtx(high3,expa3,espa3,ialx3,ialy3,chupx3,chupy3,
&                      ipath3,ipol3,iprec3,icol3)
c
call gqcntn(ierr,ntra)
call gselnt(ntr)
y=ytx
i=0
3  v=vxmin+float(i)*vxinc
if (v.gt.vxmax) goto 10
x=x0+float(i)*xinc
call cvfmt(v,fmt,strg,1)
if (l.ne.0) call gtx(x,y,strg(1:l))
8  i=i+1
if (mod(i,ninc).ne.0) goto 8
goto 3
10 call ratrtx
call gselnt(ntra)
15 return
end

```

```

subroutine valyd(ntr,vymin,vymax,vyinc,ninc,fmt,xtx)
c
c -----
c
c          Bibliothèque BBGKS
c
c -----
c          Atelier Info. Brest JJ Lechauve 1989
c
c          But:      Ecriture des valeurs sur axe vertical droit
c
c          ntr      E I4 : Numéro de la fenêtre
c          vymin   E R4 : Val. min.
c          vymax   E R4 : Val. max.
c          vyinc   S R4 : Val. increment
c          ninc    S I4 : Fréquence d'inscription
c          fmt     E C15: Format de codage des nombres
c          xtx     E R4 : Abscisse (NDC) cadrage à gauche
c
character * 15 strg,fmt
common / systm / rx,ry,rtx,rty
common / defmar / xmarg,xmard,ymarb,ymarh
common / atxtyd/ high4,expa4,espa4,ialx4,ialy4,chupx4,
1           chupy4,ipath4,ipol4,iprec4,icol4
if (vymax.lt.vymin) goto 15
if (ymarh.lt.ymarb) goto 15
y0=ymarb
y1=ymarh
dy=y1-y0
dvy=vymax-vymin
yinc=dy/dvy*vyinc
c
call patrtx(high4,expa4,espa4,ialx4,ialy4,chupx4,chupy4,
&           ipath4,ipol4,iprec4,icol4)
c
call gqcntn(ierr,ntra)
call gselnt(ntr)
x=xtx
i=0
3  v=vymin+float(i)*vyinc
if (v.gt.vymax) goto 10
y=y0+float(i)*yinc
call cvfmt(v,fmt,strg,1)
if (l.ne.0) call gtx(x,y,strg(1:1))
8  i=i+1
if (mod(i,ninc).ne.0) goto 8
goto 3
10 call ratrtx
call gselnt(ntra)
15 return
end

```

```

subroutine valyg(ntr,vymin,vymax,vyinc,ninc,fmt,xtx)
c
c -----
c
c          Bibliothèque BBGKS
c
c -----
c          Atelier Info. Brest JJ Lechauve 1989
c
c          But:      Ecriture des valeurs sur axe vertical gauche
c -----
c          ntr       E I4 : Numéro de la fenêtre
c          vymin    E R4 : Val. min.
c          vymax    E R4 : Val. max.
c          vyinc    S R4 : Val. increment
c          ninc     S I4 : Fréquence d inscription
c          fmt       E C15: Format de codage des nombres
c          xtx       E R4 : Abscisse (NDC) cadrage à droite
c
character * 15 strg,fmt
common / systm /   rx,ry,rtx,rty
common / defmar / xmarg,xmard,ymarb,ymarh
common / atxtyg/ high6,expa6,espa6,ialx6,ialy6,chupx6,
1           chupy6,ipath6,ipol6,iprec6,icol6
if (vymax.lt.vymin) goto 15
if (ymarh.lt.ymarb) goto 15
y0=ymarb
y1=ymarh
dy=y1-y0
dvv=vymax-vymin
yinc=dy/dvv*vyinc
c
call patrtx(high6,expa6,espa6,ialx6,ialy6,chupx6,chupy6,
&               ipath6,ipol6,iprec6,icol6)
c
call gqcntn(ierr,ntra)
call gselnt(ntr)
x=xtx
i=0
3  v=vymin+float(i)*vyinc
if (v.gt.vymax) goto 10
y=y0+float(i)*yinc
call cvfmt(v,fmt,strg,l)
if (l.ne.0) call gtx(x,y,strg(1:l))
8  i=i+1
if (mod(i,ninc).ne.0) goto 8
goto 3
10 call ratrtx
call gselnt(ntra)
15 return
end

```

⇒ Positionnement des attributs de polyligne

```
subroutine patrpl(itypl,wlarg,icol)
c
c -----
c
c          Bibliothèque BBGKS
c
c -----
c          Atelier Info. Brest JJ Lechauve 1989
c
c          But:      Positionnement des attributs polyline
c          ---
c          itypl      E I4 : Type de ligne
c          wlarg     E R4 : Largeur de ligne
c          icol       E I4 : Index couleur de ligne
c
c
c          Sauvegarde
c
c          call gqln(ier,itypla)
c          call gqlwsc(ier,wlarg)
c          call gpplci(ier,icola)
c
c          Positionnement
c
c          if (itypl.ne.9999) call gsln(itypl)
c          if (wlarg.ne.9999.) call gslwsc(wlarg)
c          if (icol.ne.9999) call gsplci(icol)
c          return
c
c          entry ratrpl
c
c          Sauvegarde
c
c          call gsln(itypla)
c          call gslwsc(wlarg)
c          call gsplci(icola)
c          return
c          end
```

⇒ Positionnement des attributs de polygone

```
subroutine patrfa(stylr,styli,icol)
c -----
c
c          Bibliothèque BBGKS
c
c -----
c          Atelier Info. Brest JJ Lechauve 1989
c
c          But:      Positionnement des attributs polygone (individuel)
c -----
c          stylr    E I4 : Style de remplissage
c          styli    E I4 : Style interieur (index)
c          icol     E I4 : Index couleur
c
c          integer * 4 stylr,stylra,styli,stylia
c
c          Sauvegarde
c
c          call gqfasi(ier,stylia)
c          call gqfais(ier,stylra)
c          call gqfaci(ier,icola)
c
c          Positionnement
c
c          if (stylr.ne.9999) call gsfais(stylr)
c          if (styli.ne.9999) call gsfasi(styli)
c          if (icol.ne.9999)  call gsfaci(icol)
c
c          return
c
c          entry ratrfa
c
c          Restauration
c
c          call gsfais(stylra)
c          call gsfasi(stylia)
c          call gsfaci(icola)
c
c          return
c          end
```

⇒ Positionnement des attributs de polymarque

```
subroutine patrpm(itypm,dim,icol)
c
c -----
c
c          Bibliothèque BBGKS
c
c -----
c          Atelier Info. Brest JJ Lechauve 1989
c
c          But:      Positionnement des attributs polymarque
c
c          ----
c          itypm      E I4 : Type de marque
c          dim        E R4 : Taille de la marque
c          icol       E I4 : Index couleur de marque
c
c
c          Sauvegarde
c
c          call gqmk(ier,itypma)
c          call gqmksc(ier,dima)
c          call gqpmci(ier,icola)
c
c          Positionnement
c
c          if (itype.ne.9999) call gsmk(itypm)
c          if (dim.ne.9999.) call gsmksc(dim)
c          if (icoul.ne.9999) call gspmci(icol)
c
c          return
c
c          entry ratrpm .
c
c          Restauration
c
c          call gsmk(itypma)
c          call gsmksc(dima)
c          call gspmci(icola)
c          return
c          end
```

⇒ Positionnement des attributs de texte

```
subroutine patrtx(high,expa,espa,ialx,ialy,chupx,chupy,
&                      ipath,ipol,iprec,icol)
c
c -----
c
c          Bibliothèque BBGKS
c
c -----
c          Atelier Info. Brest JJ Lechauve 1989
c
c          But:      Positionnement des attributs texte
c -----
c          high      E R4 : Hauteur (NDC)
c          expa     E R4 : Expansion ( fct.*high)
c          espa      E R4 : Espacement
c          ialx     E I4 : Alignement abscisse ( / (x,y) )
c          ialy     E I4 : Alignement ordonnée ( / (x,y) )
c          chupx    E R4 : Vecteur hauteur
c          chupy    E R4 : Vecteur base
c          ipath    E I4 : Sens d'écriture
c          ipol     E I4 : Numéro de la police
c          iprec    E I4 : Precision
c          icol     E I4 : Index couleur du texte
c
c
c          Sauvegarde
c
c          call gqchh(ier,higha)
c          call gqchxp(ier,expaa)
c          call gqchsp(ier,espa)
c          call gqtxal(ier,ialxa,ialya)
c          call gqchup(ier,chupxa,chupya)
c          call gqtxp(ier,ipatha)
c          call gqtxfp(ier,ipola,ipreca)
c          call gqtxci(ier,icola)
c
c          Positionnement
c
c          if (high .ne. 9999.) call gschh(high)
c          if (expa .ne. 9999.) call gschxp(expa)
c          if (espa .ne. 9999.) call gschsp(espa)
c          if (ialx.ne.9999.and.ialy.ne.9999) call gstxal(ialx,ialy)
c          if (chupx.ne.9999..and.chupy.ne.9999.) call gschup(chupx,chupy)
c          if (ipath .ne. 9999 ) call gtxp(ipath)
c          if (ipol.ne.9999.and.iprec.ne.9999) call gtxfp(ipol,iprec)
c          if (icol .ne. 9999 ) call gtxci(icol)
c
c          return
```

```

c      entry ratrtx
c
c      Restauration
c
c      call gschh(higha)
c      call gschxp(expaa)
c      call gschsp(espaa)
c      call gstxal(ialxa,ialya)
c      call gschup(chupxa,chupya)
c      call gstxp(ipatha)
c      call gstxfp(ipola,ipreca)
c      call gstxci(icola)
c      return
c      end

```

⇒ Gestion des chaines de caractères

```

subroutine extcha(stri,idep,nbc,stro,lstro)
c
c -----
c
c          Bibliothèque BBGKS
c
c -----
c          Atelier Info. Brest JJ Lechauve 1989
c
c          But : Extraction de la partie utile d'une chaine
c                  retournee par un Request String
c                  Le caractere de remplissage est '_'.
c
c          stri      E C80 : Chaine fournie par le request String
c          idep     E I4   : Indice du caractere debut
c          nbc      E I4   : Nombre total de caracteres de stri
c          stro     S C80 : Chaine resultat
c          lstro    S I4   : Nombre de caracteres de stro
c
c          character * 80 stri,stro
c          do 10 i=nbc,idep,-1
c          l=i
c          if (stri(l:l).ne._') goto 20
10         continue
20         if (l.eq.idep) then
            lstro=0
            stro=' '
            return
        endif
        lstro=l-idep+1
        stro(1:lstro)=stri(idep:1)
        return
    end

```

```
subroutine scanl(str,c1,c2,v,nbv,iret)
C
C -----
C
C   Retourne au programme appelant les elements d'une chaine
C   de caracteres separees par des virgules
C   En entree :
C           STR      chaine à decouper
C           C1       n° du caractere de debut de chaine
C           C2       n° du caractere de fin de chaine
C   En sortie :
C           V        tableau contenant les valeurs decodees
C           NBV     nombre de valeurs decodees
C           IRET    code d'erreur 0 --> pas de problemes
C                           -1 --> chaine incorrecte
C
C -----
C   Atelier Informatique Centre ORSTOM BREST 1988/06
C
integer c1,c2
character *1 str(1),car
character *15 temp
dimension v(1)
iret=0
nv=1
do 10 i=c1,c2
if (str(i).eq.',') nv=nv+1
10 continue
nbv=nv
temp=' '
l=15
do 20 i=c2,c1,-1
car=str(i)
if (car.eq.' ') goto 20
if (car.ne.',') temp(l:1)=car
if (car.eq.','.or.i.eq.c1) then
    read(temp,'(f15.0)',err=30) v(nv)
    nv=nv-1
    temp=' '
    l=15
    goto 20
endif
l=l-1
20 continue
goto 40
30 iret=-1
40 return
end
```

⇨ Gestion/visualisation des boites

```
subroutine ddial(ntr,xmin,xmax,ymin,ymax)
c
c -----
c
c          Bibliothèque BBGKS
c
c -----
c          Atelier Info. Brest JJ Lechauve 1989
c
c          But: Définition de la clôture de la transf. de norm.
c          associee à la zone de dialogue
c
c          ntr    E I4 : no de la transformation associee
c          xmin   E R4 : abscisse coin inf. gauche
c          xmax   E R4 : abscisse coin sup. droit
c          ymin   E R4 : ordonnée coin inf. gauche
c          ymax   E R4 : ordonnée coin sup. droit
c
c
c          common /systm/ rx,ry,rtx,rty
c          call gsvp(ntr,xmin*rtx,xmax*rtx,ymin*rty,ymax*rty)
c          return
c          end
```

```
subroutine vdial(ntr,icolf,icolb)
c
c -----
c
c          Bibliothèque BBGKS
c
c -----
c          Atelier Info. Brest JJ Lechauve 1989
c
c          But:      Videage de la zone de dialogue
c          ----
c
c          ntr        E I4 : Numéro de la fenêtre
c          icolf      E I4 : index couleur intérieur (fond)
c          icolb      E I4 : index couleur du bord
c
c          integer * 4 stylr,styli
c          call gqcntn(ierr,ntra)
c          call gselnt(ntr)
c          stylr=1
c          styli=1
c          icoli=icolf
c          icont=1
c          icolc=icolb
c          call visfen(ntr,stylr,styli,icoli,icont,icolc)
c          call gselnt(ntra)
c          return
c          end
```

```

subroutine visfe2(ntr,stylr,styli,icoli,icont,icolc)
c
c -----
c               Bibliothèque BBGKS
c -----
c Atelier Info. Brest JJ Lechauve 1989
c
c But:      Visualisation d une fenetre (remplissage)
c ----      Le contour est eventuellement materialise
c           Les bords verticaux sont des demi-cercles
c           La fenetre de la tranf. de Normalisation
c           est supposee etre (0,1)x(0,1)
c
c     ntr      E I4 : Numéro de la fenetre
c     stylr    E I4 : Style de remplissage
c     styli    E I4 : Style interieur (index)
c     icoli    E I4 : index couleur interieur
c     icont    E I4 : contour 1=oui 0=non
c     icolc    E I4 : couleur contour
c
c
real x(100),y(100),win(4),view(4)
integer * 4 stylr,stylr0,styli
call gqnt(ntr,ier,win,view)
dx=view(2)-view(1)
dy=view(4)-view(3)
c   a demi grand axe (vertical)
c   b demi petit axe (horizontal)
a=.5
b=dy/(2.*dx)
x(1)=b
y(1)=1.
x(2)=1.- b
y(2)=1.
j=2
do 10 i=1,45
teta=3.14159*(.5-float(i)/45.)
x(i+j)=x(2)+b*cos(teta)
y(i+j)=a*a*sin(teta)
10 continue
x(48)=x(1)
y(48)=0.
j=48
do 15 i=1,45
teta=3.14159*(1.5-float(i)/45.)
x(i+j)=x(1)+b*cos(teta)
y(i+j)=a*a*sin(teta)
15 continue
call gqcntn(ierr,ntra)
call gselnt(ntr)
call patrfa(stylr,styli,icoli)
call gfa(93,x,y)
call ratrfa
if (icont.eq.1) then
  stylr0=0
  call patrfa(stylr0,styli,icolc)
  call gfa(93,x,y)
  call ratrfa
endif
call gselnt(ntra)
return
end

```

⇒ Détermination d'une zone d'écho

```
subroutine zecho(ntrn,xmin,xmax,ymin,ymax,x1,x2,y1,y2)
c
c -----
c
c          Bibliothèque BBGKS
c
c -----
c          Atelier Info. Brest JJ Lechauve 1989
c
c          But:      Calcul de la zone d'echo (coord. DC)
c -----
c
c          ntrn      E I4 : Numéro de la transf. norm.
c          xmin      E R4 : Abscisse gauche WC
c          xmax      E R4 : Abscisse droite WC
c          ymin      E R4 : Ordonnée basse WC
c          ymax      E R4 : Ordonnée haute WC
c          x1        S R4 : Abscisse gauche DC
c          x2        S R4 : Abscisse droite DC
c          y1        S R4 : Ordonnée basse DC
c          y2        S R4 : Ordonnée haute DC
c
c          dimension win(4),view(4)
c          common / systm / rx,ry,rtx,rty
c          call gqnt(ntrn,ierr,win,view)
c          r1=(view(2)-view(1))/(win(2)-win(1))
c          r2=(view(4)-view(3))/(win(4)-win(3))
c          s=amax0(rx,ry)
c          x1=((xmin-win(1))*r1+view(1))*s
c          x2=((xmax-win(1))*r1+view(1))*s
c          y1=((ymin-win(3))*r2+view(3))*s
c          y2=((ymax-win(3))*r2+view(3))*s
c          return
c          end
```

```
subroutine zectxt(ntr,lnch,nblig,x1,x2,y1,y2)
c
c -----
c
c          Bibliothèque BBGKS
c
c -----
c          Atelier Info. Brest JJ Lechauve 1989
c
c          But : Definition de la zone d'echo d'un request string
c
c
c          ntr   E I4 : no de la transformation associee
c          lnch  E I4 : longueur de la chaine a gerer
c          nblig E I4 : nombre de lignes de la zone d echo
c          x1    S R4 : abscisse coin inf. gauche (DC)
c          x2    S R4 : abscisse coin sup. droit  (DC)
c          y1    S R4 : ordonnee coin inf. gauche (DC)
c          y2    S R4 : ordonnee coin sup. droit  (DC)
c
c          on suppose l'aire d'affichage (totale)
c          etre un ensemble de 24 lignes de 80 colones.
c
c          real win(4),view(4)
c          common /systm/ rx,ry,rtx,rty
c
c          call gqnt(ntr,ier,win,view)
c          s=amax0(rx,ry)
c          x1=float(nint(view(1)*s))
c          y1=float(nint(view(3)*s))
c          chx=rx/80.
c          chy=ry/24.
c          x1=x1+chx/2.
c          y1=y1+chy/2.
c          x2=x1+float(lnch)*chx
c          y2=y1+float(nblig)*chy
c          return
c          end
```

⇨ Gestion des erreurs d'entrée/sortie

subroutine pbes

```
c
c -----
c          Bibliotheque BBGKS
c -----
c          Atelier Info. Brest JJ Lechauve 1989
c
c          But: Arret d'urgence, probleme entree/sortie fichier
c -----
c          codop      E C10 : Type d'entree/sortie
c          namf      E C20 : Nom du fichier
c          ios        E I4  : Code bibli. Fortran de l'erreur
c
c
c          character * 10 codop
c          character * 20 namf
c          integer * 4 ios
c          common /essos / ios,namf,codop
c
c          call geclks
c          write (*,10) codop,namf,ios
10       format(//' Erreur en ',A10,' sur fichier ',a20,' code :,i5//)
          stop 'abort'
          end
```

