Notice d'utilisation `xjpdraw`, version 4.2

6 Octobre 1995

Jean-Pierre MERLET

Résumé

Nous décrivons l'utilisation d'un éditeur de dessin fonctionnant sur n'importe quelle station de travail dans l'environnement X. Cet éditeur génére des fichiers au format LaTeX pour les textes et au format Postscript pour les dessins qui peuvent être inclus dans un source LaTeX car ils sont englobés dans un environnement `picture`.

Il permet la création aisée de primitives géométriques variées (lignes, textes, rectangles, polygones, ellipses, ovales, courbes quelconques, 3D) et fournit de nombreux outils pour leurs manipulations.

De plus ce logiciel dispose de deux langages permettant de décrire un dessin dans un fichier externe, ce qui permet une aide commode pour la création de dessins comme résultat de programmes divers (par exemple CAO, traitement d'image).

Chapitre 1

Principes de base

1.1 Introduction

Avant de procéder à l'utilisation de `xjpdraw` il est nécessaire de procéder à une installation correcte. Pour cela consulter le chapitre "Installation". En deux mots il est nécessaire d'indiquer à `xjpdraw` une configuration de fonctionnement, ce qui est fait en indiquant dans la variable d'environnement `JPDRAW` le nom et la description d'accès à un fichier appelé le `configurateur`. Pour créer un configurateur consulter le chapitre "Installation", pour l'utiliser ajuster votre variable `JPDRAW`.

Vous n'aurez sans doute pas besoin de lire cette notice complètement avant un certain temps de pratique. Mais la première chose à faire est de lire complètement ce chapitre devant le programme actif. Dans cette configuration vous serez sûr d'être dans la bonne situation de départ et vous pourrez vous entraîner à la manipulation des différents éléments du programme.

Toutes remarques ou suggestions est à envoyer à l'adresse électronique suivante:

Jean-Pierre.Merlet@sophia.inria.fr

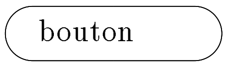
`xjpdraw` est un outil de dessin orienté géométrie fonctionnant dans l'environnement X-Window. Il est même possible d'utiliser dans certains cas `xjpdraw` avec un simple terminal non graphique. Il permet de tracer des figures simples et génére des fichiers pour l'éditeur de texte LaTeX (en utilisant l'environnement Picture, page 101 du Manuel LaTeX).

Un dessin `xjpdraw` est constitué de plusieurs éléments appelés **"primitives"**. Ce sont des lignes, des textes, des carrés etc...Le but de cet éditeur est de manipuler ces primitives à l'aide de la souris et de quelques touches fonction. Le nombre maximum de primitives est de 1500. Ce nombre peut toutefois être ajusté, voir page 100.

1.2 Boutons, Menus et Icônes

1.2.1 Boutons et menus

Ce programme fait particulièrement usage de deux sortes de représentations graphiques pour permettre à un utilisateur de choisir le type de travail qu'il désire exécuter. Il s'agit tout d'abord des *boutons* qui se présentent sous la forme suivante:



Pour valider le choix indiqué par le bouton il faut tout d'abord placer la souris à l'intérieur de l'ovale (dont la bordure extérieure change alors d'épaisseur). Lorsque on en est là deux options se présentent: soit un menu est associé au bouton, soit le bouton ne permet qu'une action unique. Un menu associé à un bouton va apparaître en appuyant sur le bouton de droite de la souris et en le maintenant appuyé.

Ce menu se compose de différentes cases disposées selon un axe vertical indiquant les choix possibles pour ce bouton. Si aucun choix n'apparaît il suffit alors de cliquer avec le bouton de gauche de la souris pour exécuter la fonction unique associée au bouton.

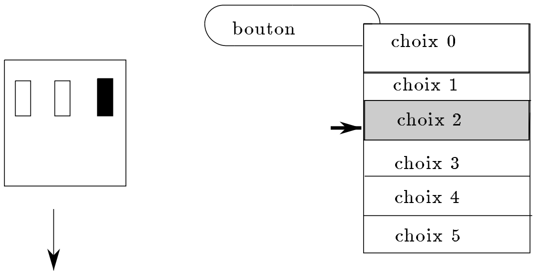


FIG. 1.1 - *Choix dans un menu*

Si un menu complet apparaît on peut sélectionner le choix désiré en placant la flêche de la souris à l'intérieur de la case correspondante tout en maintenant le bouton de droite de la souris appuyé. A chaque passage dans une case choix la case devient d'une couleur complémentaire par rapport aux autres cases (figure 1.1).

Pour valider la sélection il suffit de relacher le bouton de droite. Noter que si dans la manipulation des menus vous utilisez le bouton du milieu à la place du bouton de droite la commande ne sera pas effectuée mais vous obtiendrez un court commentaire sur ce que fait la commande.

Dans certains boutons, les fenêtres de choix, les choix du bouton sont présentés en permanence: il faut alors cliquer dans le bouton pour valider un choix.

Remarques: Après avoir fait apparaître le menu d'un bouton le déplacement de la souris ne sert qu'a sélectionner un choix dans ce menu. Par conséquent on ne peut pas activer le menu d'un autre bouton sans relacher le bouton de droite de la souris.

1.2.2 Icônes

1.2.2.1 Principe

Fondamentalement les icônes sont identiques dans leur manipulation aux boutons avec ou sans menu. La seule différence est que les différents choix possibles et la présentation même de l'icône sont représentés non pas par des textes mais par des dessins et que les choix apparaissent soit en cliquant dans l'icône successivement, soit après action sur une touche. Les icônes dans `xjpdraw` passent en négatif lorsqu'elles sont sélectionnées.

1.2.2.2 Exemple: Quitter

En cliquant sur cette icône vous sortez de `xjpdraw`. On demandera confirmation de la sortie et on signale si le fichier a été modifié sans être sauvé.

1.2.3 Les fenêtres de `xjpdraw`

Dans `xjpdraw` on utilise une fenêtre particulière: la fenêtre de texte. Ce sont des fenêtres qui permettent un dialogue entre le programme et l'utilisateur (par exemple pour donner le nom d'un fichier, une valeur etc..). Ces fenêtres ne sont actives que si la souris est placée à l'intérieur de la fenêtre. Pour transmettre le texte au programme il faut en général taper la chaîne de caractères suivie d'un retour chariot. Pour annuler la commande il suffit en général de taper seulement un retour chariot sur une chaîne vide.

Ayant défini les éléments qui vont permettre de contrôler les actions de `xjpdraw` on peut maintenant lancer le programme.

1.3 Lancer le programme

Le programme est invoqué par la commande `xjpdraw`. (cette commande est paramètrable, comme nous le verrons dans la section page 99) Si vous n'arrivez pas à lancer `xjpdraw` consulter votre gourou local ou, si *vous* êtes le gourou local, allez consulter le chapitre "Installation".

1.3.1 Tableau de bord

Après avoir lancé la commande `xjpdraw` apparaît un graphique composé de divers éléments (Figure 1.2):

1. un tableau de menus (en haut, horizontal)
2. un tableau d'icônes (à gauche, vertical)
3. un canevas (où seront tracées les figures)
4. une mini-fenêtre de messages où seront imprimés les messages émis par `xjpdraw` pour indiquer les possibilités de commandes ou son état final (en bas horizontal)
5. une mini-fenêtre contenant un éditeur de texte de type `emacs` (en bas horizontal)
6. une mini-fenêtre "Cursor" où sera indiquée la position courante de la souris (en bas horizontal)
7. une mini-fenêtre donnant le nombre de primitives existantes (en bas horizontal)
8. une fenêtre de texte, dite fenêtre des fichiers servant à communiquer avec `xjpdraw` (en bas horizontal)
9. une mini-fenêtre ("fast info") donnant des indications sur ce que fait actuellement `xjpdraw` ou ce qu'il attend que vous fassiez (en haut, à droite).
10. une mini-fenêtre ("aimant") dont nous décrirons le rôle ultérieurement.
11. une mini-fenêtre indiquant le numéro de la page et le nombre total de page. Cette fenêtre n'apparaîtra qu'en mode transparent, que nous traiterons dans un chapitre particulier.

Attention il est nécessaire que les fenêtres "fast info" et "aimant" soient totalement en dehors de la fenêtre principale de xjpdraw. Le programme testera d'ailleurs cette géométrie et émettra un message d'erreur si ces fenêtres ne sont pas positionnées correctement.

On peut sélectionner une icône en cliquant dedans mais pour certaines d'entre elles il existe des raccourcis basés sur l'usage des touches fonctions de type F lorsque la souris est dans le canevas. Le raccourci est indiqué dans l'icône. Par exemple pour tracer des segments de droite on utilise la touche F3.

Le canevas comporte des "glisseurs" qui permettent de visualiser les différentes parties du dessin (normalement depuis la version 4.1 l'utilisation des glisseurs n'impose pas de faire redessiner le dessin). La mini-fenêtre de messages va afficher les informations sur les commandes, les erreurs de manipulation. Celle contenant l'éditeur `emacs` va permettre d'éditer les primitives de texte.

1.3.2 Géométrie

Le placement des icônes et des différents tableaux, qui est indiqué dans la figure 1.2, dépend de la géométrie associée à l'exécutable `xjpdraw`. La ressource `Geometry` dans votre fichier `.Xdefaults` permet de modifier cette géométrie. Une valeur standard pour les stations SUN est définie par:

```
xjpdraw.Geometry : 1000x870+0+0
```

Il est important de bien positionner la fenêtre `xjpdraw` tout à gauche de l'écran. En effet certaines fenêtres annexes vont venir se placer tout à droite et certaines fonctionnalités de `xjpdraw` ne fonctionneront que si ces fenêtres ne recouvrent pas la fenêtre principale.

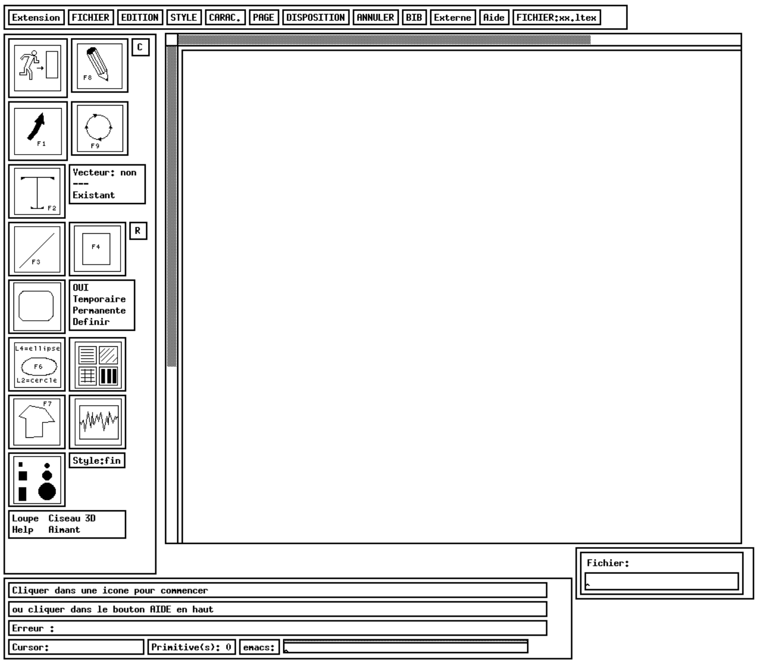


FIG. 1.2 - *Tableau de bord*

1.3.3 Nom des touches

De nombreuses touches non alpha-numériques seront utilisées dans `xjpdraw` pour indiquer différents modes de fonctionnement. Dans cette notice nous utiliserons pour les désigner des noms qui correspondent à certains claviers de station de travail SUN, c'est-à-dire une lettre (R,L,F) suivie d'un nombre.

Ces commandes peuvent être en fait assignées à des touches quelconques (voir le chapitre "Installation") et leur nom réel sera indiqué dans les informations et dans la documentation en ligne de `xjpdraw`.

1.4 Le tracé de primitives

Le choix du type de primitives que l'on va tracer s'effectue à l'aide des icônes. Le tracé d'une primitive peut parfois gommer partiellement une primitive déjà existante: pour voir à nouveau le dessin dans son état correct il suffit d'utiliser la commande `Redessiner Tout` dans le menu `PAGE`. Noter que depuis la version 4.0 cette éventualité se produit de moins en moins souvent. Mais c'est tout de même une bonne introduction aux *raccourcis*. Comme la commande `Redessiner Tout` est très souvent utilisée on peut la faire exécuter sans avoir à la chercher dans le menu: pour cela il suffit d'avoir la souris dans le canevas et d'appuyer sur la touche F5. Beaucoup de commandes de `xjpdraw` admettent de tels raccourcis. Remarquez aussi que des informations d'aide apparaissent à chaque fois que l'on clique dans une icône. Ce sont des informations générales sur les possibilités de commande dans le mode où vous entrez. A partir du moment où vous avez commencé une commande la fenêtre "fast info" vous indique à tout moment l'action suivante ainsi qu'éventuellement les caractéristiques de la primitive en cours de création. Pour ce qui concerne les menus une description sommaire des commandes est obtenu en ouvrant le menu et en le parcourant en utilisant le bouton du milieu de la souris.

Le but des sections suivantes est de décrire la manière dont on va sauver le dessin dans un fichier. Si vous voulez simplement créer des primitives (pour un premier test) reportez vous à la page 13.

1.5 Comment sauver dans un fichier

A ce stade on suppose que vous avez créer des primitives (la manière de faire est décrite dans le chapitre suivant). Nous allons maintenant sauver le dessin crée et l'imprimer. Pour cela nous allons utiliser une instruction du menu

`FICHIER`.
Préalablement il est nécessaire de savoir que `xjpdraw` peut générer le dessin sous deux formes générales différentes:

1. en utilisant les possibilités graphiques très limitées de LaTeX. Deux fichiers sont crées.
2. en sauvant directement le dessin en Postscript, sauf le texte qui est sauvé en LaTeX. C'est évidemment la manière la plus efficace d'obtenir un dessin. Il y a deux variantes dans ce cas: deux ou trois fichiers sont crées.

Dans tous les cas de figures pour l'utilisateur un dessin s'inclut dans un source LaTeX de manière unique. **Il faut recommander l'utilisation systématique de la deuxième manière.**

1.5.1 Les fichiers générés

`xjpdraw` génére toujours au moins deux types de fichier avec les extensions `.ltex` (usuellement) et `.macdraw` (par défaut `xx.ltex` et `xx.macdraw`)

– Les fichiers avec l'extension `.ltex` contiennent la description du dessin en LaTeX, c'est-à-dire un environnement `picture` que l'on peut introduire directement par la suite dans son fichier de texte LaTeX. Si l'on sauve le fichier en utilisant la deuxième méthode il y a deux alternatives:

  1. soit un fichier Postscript d'extension `.ps` sera crée (par défaut `xx.ps`)
  2. ou le Postscript sera inclu dans le fichier `.ltex`: dans ce cas le fichier `.ps` n'est évidemment pas crée.

– Les fichiers `.macdraw` sont des fichiers spéciaux qui permettent de recharger par la suite une figure faite par `xjpdraw`. Ce sont les seuls que `xjpdraw` peut relire.

Nous allons décrire maintenant comme l'on sauve dans le mode recommendé.

1.5.2 Sauver dans le fichier par défaut

Vous avez généré un dessin qui est présent dans le canevas. Placez maintenant la souris dans le bouton `FICHIER` et appuyer sur le bouton de droite de la souris. Vous allez voir apparaître un menu dont un élément est:

`Sauver le dessin en Postscript et le Texte en LaTeX`

Placer votre souris dans ce sous-choix; celui-ci devient alors noire. Relacher le bouton de la souris. Vous allez alors voir apparaître dans la mini-fenêtre d'information un message indiquant que votre fichier a été sauvé, sortez alors de `xjpdraw`. Vous allez alors constater la présence dans le répertoire courant de trois fichiers de nom `xx.ltex`, `xx.ps` (ou `xx.ps.Z`) et `xx.macdraw`. Ces fichiers contiennent la description LaTeX, Postscript et `xjpdraw` de votre dessin. Pour en être convaincu tapez la commande `xjpdraw xx.macdraw`. Vous allez voir apparaître à nouveau votre dessin.

Une autre possibilité est d'utiliser l'instruction `Sauver PS et LaTeX dans le fichier`. Dans ce cas seuls seront sauvés les fichiers `xx.ltex` et `xx.macdraw` car le Postscript est directement inclus dans le fichier LaTeX.

1.5.3 Sauver dans un fichier donné

On va maintenant créer les fichiers mais sous un autre nom que "xx", par exemple les fichiers `toto.ltex`, `toto.ps` et `toto.macdraw`. Pour commencer assurez vous que vous n'avez pas de fichier `toto.ltex` existant dans le répertoire où vous lancez `xjpdraw`.

Pour cette opération on utilise l'instruction:

`Sauver le dessin en Postscript et le Texte en LaTeX Sous`

du menu `FICHIER`. Effectuer maintenant la même manipulation que précédemment en s'arrétant dans cette case. Mettre la souris dans la fenêtre de texte (elle s'y met d'ailleurs toute seule..) et introduisez le nom complet du fichier LaTeX que vous voulez créer (ici `toto.ltex`) (toutes les extensions sont permises sauf `.macdraw` et `.ps`), puis taper `Return`. `xjpdraw` va alors générer les trois fichiers `toto.ltex`, `toto.ps` et `toto.macdraw`. Vous allez retrouver dans le fichier `toto.ltex` la description LaTeX de votre dessin. Le nom du fichier courant (d'extension `.ltex`) apparaît dans la bande noire du cadre `xjpdraw` (si son nom est trop long il ne pourra pas être entièrement affiché: pour connaître le nom du fichier courant vous pouvez utiliser la commande `CTRL-f` qui fait apparaître une nouvelle fenêtre avec le nom complet, ainsi que le nom du répertoire où sont sauvés les fichiers nouvellement crées. Pour faire disparaître cette fenêtre appuyer à nouveau sur `CTRL-f`.).

Remarque:

Je déconseille la présence de plusieurs extensions dans les noms de fichier. L'analyse des noms est très simple et risque d'être perturbée par la présence de plusieurs "." dans le nom.

Une autre possibilité est d'utiliser l'instruction `Sauver PS et LaTeX dans un fichier sous`. Dans ce cas seuls seront sauvés les fichiers `toto.ltex` (qui contiendra en fait ce qui est dans le `toto.ps` du cas précédent) et `toto.macdraw`.

1.5.4 Ecrire dans un fichier existant

A ce stade vous avez donc un fichier `toto.macdraw`. Que se passe t'il si vous tentez d'écrire un nouveau dessin sous ce nom? Tentez l'expérience en lancant la commande `xjpdraw toto.macdraw` et en sauvant sous le nom `toto.ltex`. Vous aurez alors une sonnerie et une fenêtre de choix va apparaître. Vous pourrez alors continuer la sauvegarde ou l'annuler.

Remarque:

La date de création du fichier, le nom du créateur ainsi que le répertoire de création apparaissent en commentaire dans les fichiers `.ltex` générés.

1.6 Sauvegarde automatique

Une sauvegarde automatique du dessin est effectuée régulièrement. Si le nom du fichier courant est `toto` on sauve le fichier de sauvegarde au format `.macdraw` sous le nom `#Ctoto.macdraw` . Le temps entre les sauvegardes est ajusté automatiquement en fonction du nombre de primitives. Toutefois on peut le fixer à l'aide de la commande d'un menu (voir page 50). Si vous voulez simplement revenir en arrière dans la création vous pouvez aussi utiliser le mécanisme d'annulation (voir le bouton `Annuler` page 53).

1.7 Aide en ligne

Le bouton `Aide` dans le tableau des menus permet d'obtenir une aide en ligne. Après avoir cliqué dans ce bouton vous verrez apparaître dans une fenêtre différents chapitres de l'aide en ligne (un raccourci pour obtenir cette fenêtre est `CTRL-z`). Le premier bouton (`Fin`) permet de sortir de l'aide en ligne. En cliquant dans les autres vous ferez apparaître une fenêtre contenant le chapitre en question. Les commandes classiques d'emacs permettent d'évoluer dans un chapitre de l'aide (`CTRL-n,p`: ligne suivante et précédente, `ESC-v` page suivante, `CTRL-v`: page précédente).

L'aide la plus générale est obtenue avec le bouton `A Propos` qui permet d'avoir accès à l'ensemble de la documentation. Après avoir cliqué dans ce bouton vous pouvez faire une recherche par mots en tapant `CTRL-s`. En tapant un mot ou une phrase dans la fenêtre qui apparaît vous pourrez chercher toutes les occurences de la phrase dans la documentation en ligne.

Il est aussi possible de visualiser le fichier Postscript du manuel complet en cliquant dans le bouton `Manual`, ceci dépendant de votre configurateur et de l'installation locale (voir le chapitre "Installation").

1.8 Charger un fichier

1.8.1 A l'intérieur de `xjpdraw`

La commande `Charger` du menu `FICHIER` permet de charger un fichier au format `macdraw`. Le dessin courant est remplacé par le dessin contenu dans le fichier chargé.

Différentes fenêtres décrivant les fichiers macdraw disponibles, ainsi que les répertoires présents vont apparaître. On peut alors sélectionner un fichier à charger en cliquant dans son nom.

On peut aussi indiquer le nom du fichier à charger en l'indiquant dans la fenêtre des fichiers. Pour cela on peut soit indiquer un nom complet en omettant l'extension `.macdraw` si la chaîne de caractères ne contient pas d'autre extension, en terminant par un `Return`. On peut aussi indiquer un début de nom et demander la complétion en terminant par un espace ou un caractère de tabulation. Dans ce cas la fenêtre des fichiers montrera tous les fichiers dont les noms commencent par la chaîne donnée. On peut alors cliquer dans un de ces fichiers pour le charger. A noter que la fenêtre des fichiers ne disparait que si l'on clique dans son bouton `Quit`. Dans ce mode si la fenêtre clignote et que la sonnerie retentit c'est que le fichier demandé n'existe pas.

`xjpdraw` lit aussi les fichiers `.macdraw` qui ont été compressés par `compress` (extension .Z) ou `gunzip` (extension .gz). Pour une description plus complète de l'instruction `Charger` reportez vous page 39. Attention un fichier crée en mode transparent (voir le chapitre correspondant) ne pourra pas être chargé si `xjpdraw` n'est pas en mode transparent.

1.8.2 Au lancement du programme

Au moment du lancement du programme on peut donner comme argument à `xjpdraw` un nom de fichier `.macdraw`. Là aussi l'extension `.macdraw` n'est nécessaire que si le nom contient plusieurs extensions. Ainsi les ordres valides pour charger un fichier `toto.macdraw` sont:

```
xjpdraw toto                xjpdraw toto.macdraw
```

1.8.3 Album

La commande `Album` du menu `FICHIER` permet de rajouter au dessin courant le dessin contenu dans un fichier `.macdraw` déjà existant. Après avoir sélectionné l'album différentes fenêtres décrivant les fichiers macdraw disponibles, ainsi que les répertoires présents vont apparaître. On peut alors sélectionner un fichier à inclure en cliquant dans son nom.

On peut aussi indiquer le nom du fichier à récupérer dans la fenêtre des fichiers. Pour cela on peut soit indiquer un nom complet en omettant l'extension `.macdraw` si la chaîne de caractères ne contient pas d'autre extension, en terminant par un `Return`. On peut aussi indiquer un début de nom et demander la complétion en terminant par un espace ou un caractère de tabulation. Dans ce cas la fenêtre des fichiers montrera tous les fichiers dont les noms commencent par la chaîne donnée. On peut alors cliquer dans un de ces fichiers pour le charger.

Les primitives de l'album sont associées (voir page 26) pour que l'on puisse les déplacer en bloc facilement. Il est donc prudent de dissocier les primitives existantes préalablement.

Les fichiers indiqués peuvent avoir été compressés par `compress` (extension .Z) ou `gunzip` (extension .gz). Il n'est pas nécessaire de donner ces extensions.

1.9 Sauvegarde de l'état

Si vous définissez une variable d'environnement `JPDRAW_STATE` comme étant `ON` et que vous quittez normalement `xjpdraw` il y a création d'un fichier (*.jpdraw_state*) décrivant l'état courant de `xjpdraw`, de manière assez complète. Si vous relancez `xjpdraw` vous verrez alors le dessin dans l'état où vous l'aviez laissé. Pour inhiber cette action il suffit de mettre la variable `JPDRAW_STATE` à `OFF`. Une autre manière de sauver l'état de `xjpdraw` est de sauvegarder le contexte, voir page 43.

1.10 Imprimer

1.10.1 Les possibilités graphiques de LaTeX

Le fichier LaTeX crée par `xjpdraw` contient simplement un environnement LaTeX de type `picture` (le fichier crée par `xjpdraw` n'est donc pas compilable directement mais doit être inclus dans un source déjà existant). LaTeX permet dans cet environnement certains ordres graphiques mais très limités par rapport aux possibilités des imprimantes. En effet dans la version 2.09 on ne peut définir que:

- des rectangles (noirci ou non)
- des cercles (de rayon limité, noircis ou non)
- des textes
- des lignes ou des vecteurs (dont la pente doit être définie comme le rapport de deux entiers compris entre $\pm$ 6 ($\pm$4 pour les vecteurs) et premier entre eux)

Ainsi le fichier suivant (produit par `xjpdraw`) donne le résultat de la figure 1.3.

```
\begin{picture}(   3.0,    1.9)( 0.913,11.339)
\thinlines
\large
\put(  2.48, 11.98){\circle{  1.24}}
\put(  3.12, 11.34){\framebox(  0.81,  1.90){}}
\put(  0.91, 11.51){\line(5,4){  2.19}}
\end{picture}
```

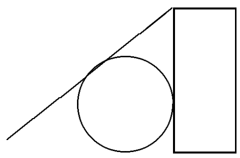


FIG. 1.3 - *Un dessin en LaTeX*

Décrire un dessin avec ce type d'ordre est relativement pénible mais est possible avec `xjpdraw`. Si vous désirez créer un fichier contenant l'ensemble de votre dessin (et qui sera donc portable sur toute machine connaissant LaTeX) il faut préalablement cliquer sur le bouton `Extensions` du tableau des menus pour faire apparaître le mot `Standard`. Dans ce mode `xjpdraw` ne permet la création que de primitives connues de LaTeX. Il faudra ensuite sauver ce fichier avec l'instruction `Sauver` ou `Sauver sous` du menu `Fichier`.

Dans les versions précédentes de `xjpdraw` était distribué un fichier décrivant des commandes TeX permettant d'étendre les possibilités de LaTeX. Ces extensions ne sont maintenant plus utilisées sauf si vous avez des textes tournés dans votre dessin (et même dans ce cas les extensions ne seront pas nécessaire si vous utilisez LaTeX 2e et le package `rotating`). L'installation des extensions est décrite dans le chapitre "Installation".

1.10.2 Incorporez votre dessin

Le fichier LaTeX tel qu'il est crée par `xjpdraw` est insérable tel quel dans le fichier source avec votre éditeur de texte. Une autre possibilité (plus propre) est d'utiliser l'instuction LaTeX `\input` comme par exemple `\input{toto.ltex}` qui prend le fichier `toto.ltex` et le met dans le source. Si vous ne mettez pas d'extension dans l'argument de l'`input`, LaTeX prend automatiquement l'extension .tex.

Si vous désirer utiliser le dessin crée comme figure il est prudent de placer un `\begin{center}` avant le input et un `\end{center}` après.

1.10.3 Les dimensions

Les ordres graphiques LaTeX utilisent une unité arbitraire de dimension. On définit la relation entre centimètres et cette unité arbitraire à l'aide de la variable `unitlength`. Cela se fait en placant l'instruction `\unitlength x cm` préalablement au dessin. Les dessins crées par `xjpdraw` sont prévus pour un `unitlength` de 1cm (il n'est d'ailleurs pas nécessaire de spécifier la valeur de `unitlength`). Changez l'unité de longueur à l'intérieur de `xjpdraw` permet de faire un agrandi ou une réduction de votre dessin (sauf pour les textes). Nous verrons que cette possibilité peut aussi être obtenue postérieurement à la création du dessin, ceci sans modification des fichiers crées par `xjpdraw`.

1.10.4 Imprimer

Il est parfois souhaité, en cours de création d'un dessin, d'imprimer ce dessin seul sur une page. La commande `Vérification LaTeX` du menu `FICHIER` permet justement de créer automatiquement un fichier de nom `bidon.ltex` qui contient un préambule LaTeX , le fichier LaTeX de description du dessin et une fin de document, ceci pour les dessins exécutés en mode `Standard`. Par défaut `xjpdraw` utilise des préambules et des fins de document très simples, qui sont simplement dépendantes de la version de LaTeX que vous utilisez, ce qui est décrit par la command `LATEX_VERSION` du configurateur. Toutefois si vous désirez avoir des préambules et des fins de document plus sophistiqués il suffira de les placer sous les noms `header_latex`, `header_end.ltex` dans le répertoire `LIB` du configurateur (voir le chapitre "Installation").

La commande `Vérification LaTeX avec Postscript` est celle pour laquelle les primitives graphiques du dessin sont en Postscript. C'est évidemment celle-ci qui sera utilisée en priorité.

Une fois ces opérations terminées il ne reste plus qu'à compiler le fichier `bidon.ltex`. Après compilation il faudra convertir le fichier .dvi en Postscript en utilisant le programme approprié (comme par exemple `dvips`), puis envoyer ce fichier sur votre imprimante. L'ensemble de ces opérations peut être effectué directement à partir de `xjpdraw` en utilisant la commande `Compiler et imprimer` du menu `FICHIER`. L'utilisation de cette commande nécessite cependant d'avoir effectué correctement l'installation. Une autre possibilité moins coûteuse en papier est d'utiliser la commande `Compiler et previewer` du même menu.

1.11 Utilitaires

La fenêtre des utilitaires, dans le tableau des icônes, permet d'utiliser cinq utilitaires. Tout d'abord la

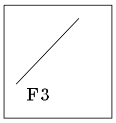
1. la loupe qui permet d'obtenir une vue agrandie d'une partie du dessin (voir page 33)
2. les ciseaux qui permettent de couper en un point une primitive (voir page 33).
3. le bouton `help` permet de faire disparaître ou apparaître le bouton de "fast info"
4. la fonction aimant permet de gérer les primitives en modifiant leurs points de création selon des règles particulières qui sont affichées dans une fenêtre (voir page 34). Le bouton `Aimant` permet de faire disparaître ou apparaître cette fenêtre.
5. le menu pour la construction de primitives 3D (voir page 20).

Chapitre 2

Création de primitives

Les primitives sont crées en activant certains modes de `xjpdraw`, ce qui se fait en cliquant dans les icônes de la fenêtre des icônes.

2.1 Ligne



Ce mode permet de tracer des lignes et des vecteurs. Si l'on veut des vecteurs cliquer avec le bouton de gauche sur le bouton vecteur: le `Non` à côté du bouton passe successivement à `Oui`, puis à "2 <>" (dans ce cas les vecteurs auront une flêche aux deux extrémités, ces flêches pointant vers l'extérieur), puis à "2><" (auquel cas on mettra deux flêches mais tournés vers l'intérieur). Pour créer un segment cliquer à l'origine du segment avec le bouton de gauche et maintenir ce bouton appuyé. En déplacant la souris le segment apparaît. Lorsque le segment voulu est obtenu relacher la souris.

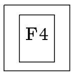
Le bouton `---` dans la fenêtre des icônes fait apparaître une fenêtre permettant de choisir le style des traits: fins, épais, pointillés fin, pointillés épais. Il permet aussi de choisir différents styles de pointillés (25 styles différents), ainsi que 5 types de tête pour les vecteurs.

Le choix fin ou épais est valable pour <u>toutes</u> les primitives qui vont être crées à partir de ce moment.

Une fenêtre de message dans le tableau des icônes indique le style courant. Nous verrons en détail page 21 comment manipuler le style des traits.

On peut forcer la pente d'une ligne à être compatible avec celle permise par le LaTeX standard (voir page 9) en maintenant la touche L8 appuyée lors de la création de la primitive. Cette opération permet par exemple de tracer des lignes parfaitement horizontales ou verticales à partir d'un déplacement approximatif de la souris.

2.2 Rectangle



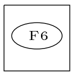
2.2.1 Principe

Ce mode permet de tracer des rectangles.

Appuyer sur le bouton de gauche de la souris pour fixer la position d'un coin du rectangle et maintenir appuyé, déplacer la souris pour fixer le coin opposé, le rectangle apparaît, puis relacher une fois la bonne position déterminée.

Pour forcer le tracé d'un carré appuyer sur L2 une fois dans le canevas (le caractère `C` apparaît en place de `R` à côté de l'icône). La longueur du côté du carré sera la plus petite des distances horizontale, verticale entre le premier coin et la position courante. Pour revenir au mode rectangle tapez sur L4.

2.3 Ellipse, Cercle, Arc



2.3.1 Ellipses

Ce mode permet de créer des ellipses. Deux modes de création sont possibles (figure 2.1):

- le centre de l'ellipse est déterminé par la position de la souris au moment où l'on clique sur le bouton de gauche pour la première fois. On maintient ce bouton appuyé et on déplace la souris. L'ellipse est contenue dans le rectangle de centre le centre de l'ellipse et dont un coin est la position actuelle de la souris.
- ellipse d'axe horizontal, vertical, passant par 4 points: pour cette fonctionnalité cliquez dans l'icône jusqu'à ce qu'apparaisse une icône avec 4 points numérotés sur l'ellipse. Cliquer en 3 points puis déplacer la souris (sans appuyer sur un bouton). Vous verrez alors apparaître une ellipse passant par les 3 points cliqués et la position courante de la souris (si l'ellipse n'apparaît pas c'est qu'il n'existe pas d'ellipse à axe horizontal passant par les 4 points). Cliquer en un quatrième point pour fixer l'ellipse.

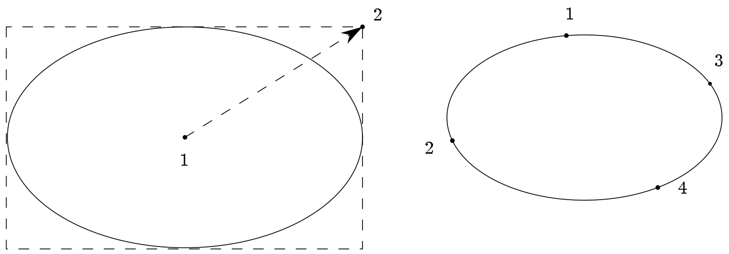


FIG. 2.1 - *Les méthodes pour créer des ellipses*

2.3.2 Cercles

L'icône des ellipses permet aussi de passer dans le mode création de cercle. Après l'avoir sélectionnée mettez la souris dans le canevas et tapez sur la touche L2. Vous verrez alors apparaître à la place de l'ellipse noir un cercle. Différentes icône avec des cercles apparaissent si l'on clique successivement dans l'icône (le bouton de gauche fait avancer dans le choix et le bouton du milieu fait reculer). Ces différentes icônes permettent de choisir la manière dont on va définir le cercle (figure 2.2).

- 1^{er} choix: cliquer avec le bouton de gauche au centre du cercle, déplacer la souris en maintenant le bouton de gauche appuyé pour indiquer le rayon du cercle, relacher le bouton une fois le bon cercle obtenu.
- 2^{eme} choix: inverse du précédent, on donne d'abord un point sur le cercle puis le centre du cercle.
- 3^{eme} choix: on donne deux points sur le cercle puis son centre. Cliquer successivement avec le bouton de gauche pour introduire les deux points sur le cercle, la médiatrice apparaît, placer la souris sur cette médiatrice pour définir le centre, puis cliquer.
- 4^{eme} choix: on donne 3 points sur le cercle en cliquant successivement aux endroits voulus avec le bouton de gauche.
- 5^{eme} choix: permet d'obtenir des cercles avec à l'intérieur deux lignes à 45 degré. Attention le cercle est donné dans le mode extérieur puis centre.

Les points à cliquer apparaissent numérotés dans l'ordre dans l'icône. Pour obtenir à nouveau les ellipses il suffit d'appuyer sur L4 (avec la souris dans le canevas).

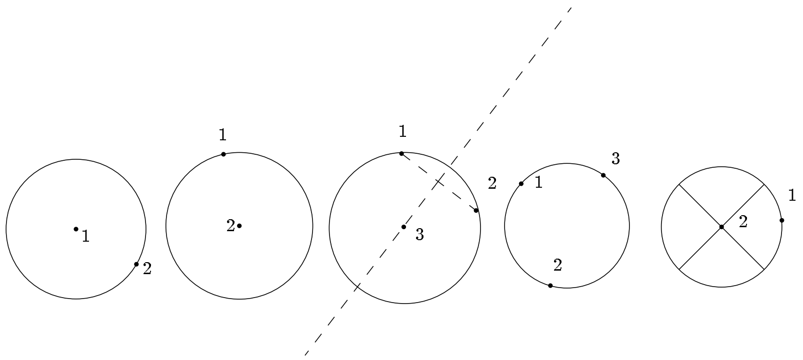


FIG. 2.2 - *Les différentes méthodes de création des cercles*

2.3.3 Arcs de cercle

On peut créer des arcs de cercle de la manière suivante. Créez tout d'abord un cercle complet en utilisant le troisième ou le quatrième menu du mode cercle (tracé avec une médiatrice ou le mode cercle passant par trois points). Une fois que le cercle est défini vous pouvez supprimer l'arc de cercle de longueur maximale entre les deux premiers points cliqués en appuyant sur la touche L6 ou son complémentaire en tapant sur L8 (figure 2.3). Notez qu'en mode vecteur ou double vecteur les arcs seront munis de flêches.

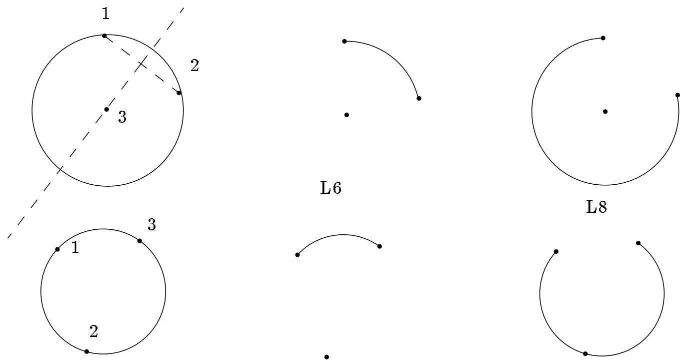
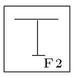


FIG. 2.3 - *Les différentes méthodes de création des arcs*

Remarque: les arcs de cercle crées de cette manière sont définis par deux angles mesurés par rapport à l'axe x. Ces angles doivent être en degré, ce qui peut avoir pour conséquence que les arcs peuvent ne pas commencer ou finir exactement aux points donnés. Une autre manière de créer des arcs est de couper un cercle aux deux points voulus à l'aide du ciseau (voir page 33).

2.4 Texte



Ce mode permet de placer un texte dans le dessin. On doit d'abord définir la taille et le style des caractères à l'aide des menus `CARAC` et `STYLE` . Par défaut la taille est 10pt et le style Roman (appelé `Standard` dans `xjpdraw`). Ces textes sont interprétés comme du LaTeX: on peut donc introduire, par exemple, des formules mathématiques complexes (voir le dessin de la page 121).

Pour introduire un texte placer votre souris à la position voulue (c'est-à-dire à l'endroit où vous voulez avoir le coin inférieur gauche du texte et il n'est pas nécessaire de cliquer à cette position) et tapez votre texte au clavier sans modifier la position de la souris. Le texte apparaît. Les caractères d'effacement sont le backspace et le delete. Une autre manière d'introduire un texte est de le sélectionner dans une autre fenêtre (par exemple dans un fichier LaTeX), de placer la souris dans le canevas et d'appuyer sur le bouton du milieu. Le texte s'affiche alors avec comme origine la position courante de la souris.

Pour terminer l'introduction d'un texte déplacez la souris. Vous le verrez apparaître dans la mini-fenêtre `emacs` et une boîte en pointillée est tracée autour du texte. Selon votre configurateur cette boîte indique:

- si l'installation a été faite de façon à permettre la compilation des textes en LaTeX(vous avez placé dans votre configurateur une commande `LATEX_COMMAND` et mis la variable `COMPILE_LATEX` à 1), la boîte indique la taille exacte qu'occupera votre texte. Pour cela il est cependant nécessaire d'attendre quelques secondes après l'entrée du texte (après introduction du texte la fenêtre "fast info" affiche le message "Compiling, wait"). Depuis la version 4.2 un processus de compilation tourne en tâche de fond en permanence ce qui permet de réduire très nettement le temps de calcul de la boite englobante, à condition toutefois de ne pas enchaîner très rapidement l'introduction de textes.

- si l'installation ne permet pas la compilation des textes ou si vous avez délibérément invalidé cette option, la boîte indique approximativement la place occupée par le texte après compilation LaTeX, l'estimation se faisant d'après le nombre de caractères.

Le choix de compiler ou non les textes se fait dans le configurateur, voir le chapitre "Installation". Pour ce qui concerne l'édition de texte après création reportez vous à la page 32.

Si vous tapez un retour chariot dans votre texte le curseur de la souris se déplace à la ligne suivante, ce qui permet de créer des textes sur plusieurs lignes.

Les lignes de texte commencant par un % sont considérées comme des commentaires pour le fichier LaTeX (ce qui est utile pour indiquer ce que représente le dessin).

Les caractères affichés à l'écran doivent correspondre approximativement au style et à la taille choisie à condition de disposer de certaines fontes particulières. Si ce n'est pas le cas il est possible d'utiliser les fontes disponibles pour obtenir le même effet (voir le chapitre "Installation" pour l'installation de vos fontes personnelles et le nom par défaut des fontes). Si vous n'avez pas les fontes qui sont utilisées par défaut dans le programme et que vous n'avez pas installé vos fontes personnelles `xjpdraw` ne fonctionnera pas. Il est nécessaire en effet qu'il dispose d'au moins quelques fontes parmi celles qui lui sont indiqués. En cas de problème utilisez le programme `xfontsel` pour déterminer les fontes disponibles et indiquez ces fontes dans le configurateur (voir le chapitre "Installation").

Si le style choisi n'apparaît pas dans le dessin, les fontes seront cependant respectées sur le papier. Bien entendu vous pouvez introduire dans le texte même des ordres de style puisque le texte sera compilé.

Il est possible d'utiliser des commandes spéciales pour les textes. En effet si `xjpdraw` doit compiler un texte il testera la présence dans le répertoire courant d'un fichier `command.latex`. Si ce fichier existe il est ajouté au fichier LaTeX généré.

Il est possible de visualiser en cours de création l'aspect définitif de votre dessin à l'aide d'un utilitaire permettant de visualiser du Postscript (voir page 42).

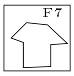
A noter le traitement spécial de certains caractères spéciaux de LaTeX qui apparaissent à l'écran sous leur forme compilé. Ce sera le cas pour la majorité des des symboles mathématiques de LaTeX. Ceci se passe de la manière suivante: pour les symboles mathématiques si vous tapez `x= $ \mu $` vous verrez alors s'afficher `x= $` μ `$`. En effet `xjpdraw` dispose de deux fontes réservées pour ces symboles spéciaux, les reconnait dans le texte et affiche le caractère correspondant. Par défaut ces fontes sont:

```
-adobe-symbol-medium-r-normal--14-140-75-75-p-85-adobe-fontspecific
-adobe-times-medium-r-normal--12-120-75-75-p-64-iso8859-1
```

mais peuvent être modifiées dans le configurateur. Cette substitution ne se fera que si le texte ne contient pas plus de 20 symboles spéciaux et si les symboles mathématiques sont séparés par des blancs. Par contre dans la fenêtre `emacs`, dans laquelle on pourra faire de l'édition, le texte original reste conservé.

Si vous translatez une primitive de texte faites attention à ne pas placer son origine hors du canevas, auquel cas il n'apparaîtra pas à l'impression.

2.5 Polygone



Ce mode permet le tracé des courbes polygonales fermées ou non.

2.5.1 Polygone général

Pour créer une ligne polygone ou un polygone fermé cliquer avec le bouton de gauche successivement aux sommets du polygone. Entre le dernier point cliqué et la position courante de la souris apparaît l'arête potentielle. Pour fermer le polygone faites un double-click (c'est-à-dire appuyer deux fois sur le bouton avec un intervalle de temps faible entre les pressions) sur le bouton du milieu. Le point où se trouve la souris est rajouté au polygone. Pour terminer une ligne polygonale non fermée cliquer avec le bouton du milieu.

Un polygone peut être annulé en cours de création par la touche R5 et on peut annuler un point cliqué par la touche R4 ou L4.

Attention: Vous ne devez pas avoir plus de 30 sommets dans un polygone! Si vous dépassez ce nombre la fenêtre clignote et la ligne générée compte le nombre maximum de point.

2.5.2 Polygones particuliers

2.5.2.1 Tête de vecteur

On peut créer un polygone qui ressemble à une tête de vecteur. Pour cela, en mode polygone tapez sur L3. Il suffit alors de cliquer sur les points indiqués dans l'icône pour faire apparaître le polygone en question. La touche L10 permet de revenir aux polygones généraux.

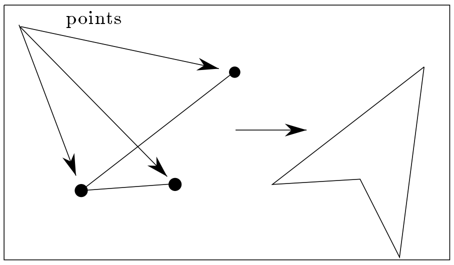
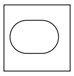


FIG. 2.4 - *Création d'une tête de vecteur*

2.5.2.2 Polygone régulier

Si vous appuyer sur la touche L9 après avoir sélectionné l'icône des polygones vous pourrez tracer des polygones réguliers (un polygone dont tous les côtés sont de longueur égale). Les deux premiers points cliqués indiquent la longueur du côté du polygone. Puis à chaque fois que vous cliquerez sur le bouton de gauche de la souris un côté s'ajoute. Ainsi en cliquant deux fois vous obtiendrez un triangle équilatéral, 3 fois un carré, 4 fois un pentagone etc....(figure 2.5) Pour terminer faite comme pour les polygones généraux. La touche L10 permet de revenir au polygones généraux.

2.6 Ovale



Ce mode permet de générer des ovales (un rectangle avec des coins arrondis) ou des parties d'ovale.

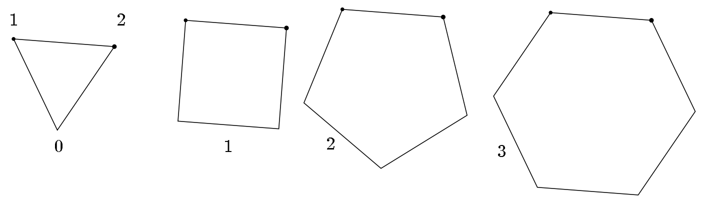


FIG. 2.5 - *Polygones réguliers: Les points 1 et 2 ont été introduit et le nombre en bas des figures indique le nombre de fois où l'on a cliqué sur le bouton de gauche de la souris après l'introduction des deux premiers points (notez qu'à chaque fois on a terminé le polygone par un double-click sur le bouton du milieu)*

En utilisant les touches L1, L2, L3, L4 va apparaître le type d'ovale voulu: on a l'ovale complet (par défaut), les 1/2 ovales et les 1/4 d'ovale et le 1/4 d'ovale avec une flêche. Les modes de création (figure 2.6) sont les suivant:

- pour l'ovale complet: cliquer pour indiquer la position du centre de l'ovale, maintenir cliqué et déplacer la souris. L'ovale apparaît. Relacher lorsque l'ovale est correct.
- pour les parties d'ovale: cliquer pour indiquer le début de la primitive et se déplacer en maintenant cliqué. Pour les 1/4 d'ovale on tourne toujours dans le sens trigonométrique a partir du point de départ. Pour les 1/2 ovales si vous êtes en mode vecteur deux flêches vont apparaître aux extrémités. Elles seront présentes dans le dessin LaTeX.

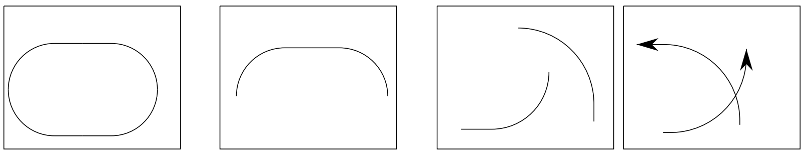
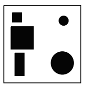


FIG. 2.6 - *Les différents types d'ovales*

2.7 Marques



Les marques sont des primitives prédéfinies que l'on peut ajouter très rapidement dans un dessin, avec la possibilité de définir ses propres marques. Un raccourci pour activer l'icône des marques est F10.

2.7.1 Les marques standards

Les marques standards disponibles sont (figure 2.7):

1. carrés noirs: dimension 2mm et 5mm
2. rectangles noirs: dimension 2x5 mm
3. cercles noirs: diamètres 0.5, 2 et 5 mm

Pour sélectionner le type de la marque cliquer successivement dans l'icône (le bouton de gauche fait avancer, celui du milieu reculer). Des icônes différentes apparaissent. Pour introduire une marque standard cliquer dans le dessin. Le point cliqué correspondra au coin inférieur gauche des carrés et au centre du cercle pour les marques circulaires.

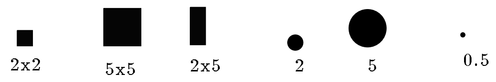


FIG. 2.7 - *Les marques standards*

Vous pouvez aussi changer de marque en tapant sur une des touches de a à h. Vous pouvez aussi tracer une marque différente de la marque courante en appuyant sur une des touches entre `A` et `H` (la première marque standard correspond à `A`).

2.7.2 Définir et utiliser une marque

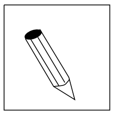
Lorsqu'un certain nombre de primitives vont être utilisées constamment dans un dessin il peut être commode de les définir comme des marques. Pour cela créer tout d'abord les primitives à utiliser puis sélectionner le choix de l'icône `marque` indiqué par un `#define` (en cliquant dans l'icône successivement). Cliquer alors sur les primitives que vous voulez définir comme des marques. A chaque marque est alors associée un chiffre entre 0 et 9, qui est d'ailleurs indiqué sur la primitive au moment où on la définit comme marque (vous ne pouvez pas définir plus de 10 marques). Pour utiliser maintenant les marques que vous avez définies sélectionner le choix de l'icône `marque` indiqué par un `#use`. Placez votre souris à l'endroit où vous voulez le centre de votre marque et tapez le chiffre associé à la marque. Si vous ne vous souvenez pas du numéro des marques tapez `Return`, le numéro des marques va apparaître sur les primitives qui ont servi à les définir (si la primitive qui a servi à définir une marque a été détruite elle sera à nouveau temporairement affichée). Pour détruire la dernière marque crée tapez '-' et pour les détruire toutes tapez '='.

Une petite astuce du mode `use`: vous pouvez l'utiliser pour mettre des marques sur certains points de la primitive sélectionnée. Pour cela:

1. définissez vos marques
2. sélectionnez une primitive en restant dans le mode `use`
3. pour mettre une marque sur la sélectionnée appuyez sur une touche de 'A' à 'F' pour les marques standardes ou '0' à n pour vos propres marques. La marque se met aux extrémités des segments, sur les sommets des rectangles et des polygones, au centre pour les ellipses, cercles et ovales.

Noter que dès que vous avez sélectionné une primitive dans le mode `use` vous ne pouvez plus utilisez les touches 'A' à 'F' et '0' à n pour mettre une marque à la position courante de la souris. Pour cela il faut annuler la sélection ce qui se fait en cliquant avec le bouton de gauche loin de toute primitive.

2.8 Le crayon



Le crayon sert à tracer différentes courbes. Il existe quatre modes de fonctionnement dans ce cas. Pour faire apparaître le menu permettant de choisir un mode il suffit d'appuyer avec le bouton de droite dans le bouton à côté de l'icône du crayon (bouton marqué `C`). Le menu apparaît avec les différentes options qui sont décrite dans les sections suivantes.

2.8.1 Dessin à la souris (`Souris`)

Ce mode est utilisé pour tracer une courbe à la main: cliquer sur le bouton de gauche et déplacer la souris en maintenant le bouton appuyé. Les positions successives de la souris sont échantillonnées et sont ensuite reliés par des segments de droite. Le nombre de point maximum est de 200. Si ce nombre de point est dépassé le tracé s'interrompt et il y a une sonnerie. C'est le mode par défaut.

2.8.2 Courbe interpolée à partir de points (`Spline`)

Dans ce cas on clique en différents points avec le bouton de gauche (en chaque point doit apparaître une croix) Pour annuler un point il suffit de cliquer sur le bouton du milieu (on ne peut pas annuler les deux premiers points). Pour finir on double-clique alors sur le bouton du milieu. La courbe passe par les points de contrôle. Après création il est possible de changer la position des points de contrôle et d'en ajouter (voir la section "Modification de taille").

2.8.3 Courbe interpolée à partir de points (`B-Spline`)

Dans le principe la création est identique à celles des splines. Ces courbes ne passent pas forcément par les points de contrôle.

2.8.4 Approximation polynomiale (`Polynôme`)

On procède de la même manière que précédemment pour introduire n points de contrôle. Le système trace alors une courbe polynomiale de degré $n-1$ passant par les points de contrôle. Les courbes polynomiales sont en général plus "raides" que les splines (voir figure 2.8).

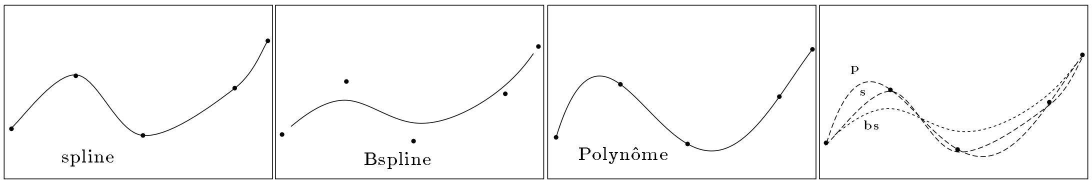


FIG. 2.8 - *Courbes Spline, B-spline et polynomiale obtenues à partir des mêmes points de contrôle.*

2.8.5 Parabole

Pour générer cette primitive (figure 2.9) on clique avec le bouton de gauche pour indiquer l'origine de la parabole et on maintient ce bouton appuyé en déplacant la souris. La pente à l'origine de la parabole sera verticale ou horizontale selon le premier déplacement de la souris. Lorsque la parabole est terminée relacher le bouton.

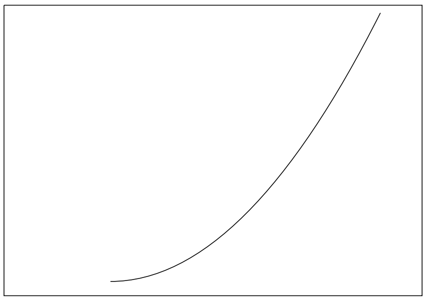


FIG. 2.9 - *Arc de parabole*

Noter que vous pouvez aussi tracer des paraboles d'axe quelconque en utilisant le mode `Formule` du menu `EDITION`, voir page 46. Vous pouvez aussi faire tourner l'arc de parabole que vous venez de créer.

2.9 Le module 3D

Dans la fenêtre des "Utilitaires" un certain nombre de possibilités de dessin 3D sont offertes. Il s'agit de création de sphère, de parallélépipède, de cylindre, de tronc de cône et d'objet de révolution. Attention le résultat n'est pas une primitive au sens `xjpdraw` mais en est simplement constituées. On ne peut donc pas les manipuler dans leur globalité après création. Après avoir sélectionné le menu 3D une nouvelle fenêtre de choix va apparaître: les sections suivantes présentent les différents choix.

2.9.1 Ajustement de la vue 3D

Le premier choix du menu 3D permet de spécifier la vue 3D sous laquelle on va dessiner les primitives. Ceci se fait à l'aide de 2 curseurs, qui définissent la direction de la ligne de vue ou bien de quatre boutons qui permettent de choisir les vues prédéfinies `Perspective` (la valeur par défaut), `Dessus`, `Face`, `Côté`. Noter qu'en cliquant dans ces boutons les glisseurs indiquent les nouvelles valeurs des angles pour la ligne de vue. Ensuite on a à choisir le type de représentation pour les surfaces à l'aide d'un bouton de choix qui permet de choisir soit `Facettes Ombrées`, `Facettes` ou `Symbolique`.

Dans le premier et le deuxième cas la surface sera représentée par des facettes polygonales. Si le mode est `Facettes Ombrées` les polygones auront une texture en niveau de gris dont l'intensité est fonction de l'angle entre la normale du polygone et la ligne de vue (gris très clair pour des lignes colinéaires). Dans le mode `Facettes` seul le contour des polygones est tracé. Le nombre de facettes est fixé à 100 par défaut pour les sphères, 10 pour les cylindres et les cônes mais peut être modifié à l'aide du curseur qui apparaît dans cette fenêtre (on y donne le nombre de facettes). Il n'y a pas de vérification de la visibilité des facettes. Dans le mode `Facettes Ombrées` il sera parfois nécessaire de procéder à un passage au premier ou second plan de certaines facettes.

Dans le mode `Symbolique` on ne représente pas la surface mais le volume 3D est illustré par la représentation symbolique classique des primitives, les arêtes cachées apparaissant en pointillé.

2.9.2 Tracé des primitives 3D

Pour tracer une primitive 3D (figure 2.10) utiliser les processus suivants:

- pour les sphères: cliquer avec le bouton de gauche de la souris au centre de la sphère puis en un autre point pour indiquer le rayon

- pour les cubes: cliquer en 4 points avec le bouton de gauche de la souris pour indiquer le centre du cube, sa demi-largeur (écart en x avec la position du centre), sa demi-hauteur (écart en y avec la position du centre), son épaisseur (distance au point précédent).

- pour les cylindres: cliquer avec le bouton de gauche de la souris en 3 points pour indiquer le centre de la base du cylindre, son rayon (écart en x avec la position de la base) puis sa hauteur (écart en y avec la position de la base).

- pour les troncs de cônes: cliquer avec le bouton de gauche de la souris en 3 points pour indiquer le centre de la base du cône, le rayon de la base (écart en x avec la position du centre), puis la hauteur (écart en y avec la position du centre) et le rayon de sa partie haute (écart en x avec la position du centre).

- objet de révolution: tracer tout d'abord une ligne polygonale et sélectionner la. Choisissez dans le menu 3D le mode objet de révolution et cliquer sur le bouton de gauche. On obtient une vue 3D de l'objet obtenu en faisant tourner la ligne polygone autour de l'axe y (figure 2.11),

2.10 Le style des traits des primitives

Le bouton `---` à côté de l'icône des lignes va faire apparaître un tableau qui permet de choisir le style de trait avec lequel on va tracer les primitives. C'est un bouton à menu avec les choix suivants:

- fin: les traits sont fins, continus

- épais: les traits sont continus, de l'épaisseur fixée par la commande `Régler l'épaisseur des traits`. Par défaut l'épaisseur des traits épais est fixée à 0.7mm.

- pointillé: les trais sont fins et en pointillé.

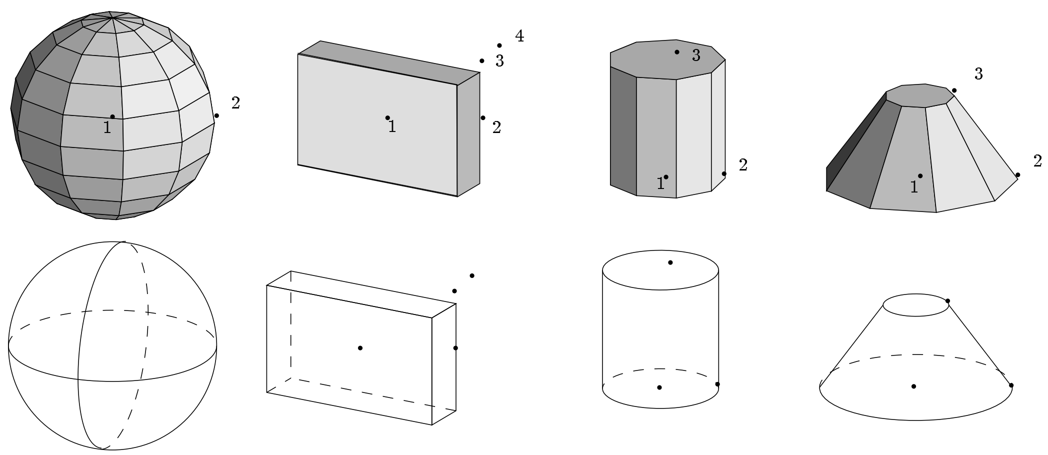


FIG. 2.10 - *Les primitives 3D tracées en facettes ombrées et en symbolique.*

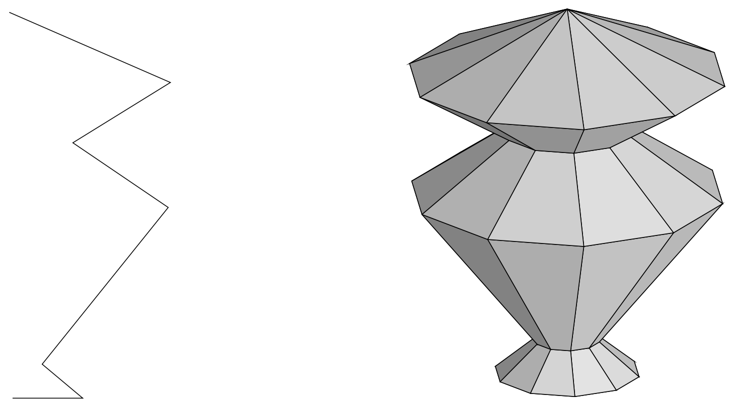


FIG. 2.11 - *Création d'un objet de révolution.*

– pointillé épais: les trais sont pointillés, de l'épaisseur fixée par la commande `Régler l'épaisseur des traits` Au départ le style épais est fixé à une épaisseur d'environ 0.7mm.

– Autres pointillés: permet de choisir un des 25 styles de pointillé (figure 2.12). En utilisant cette commande on fait apparaître une fenêtre où les différents pointillés sont représentés, lorsque la souris est dans la fenêtre. Si l'on place la souris aux environs d'un pointillé il clignote. Si on clique ensuite sur le bouton de gauche le style de pointillé est choisie et la fenêtre disparaît. `xjpdraw` passe alors automatiquement en mode pointillé fin. Le style des pointillés sera le même si l'on passe en mode pointillé épais. Un raccourci pour faire afficher la fenêtre des pointillés est `CTRL-l`.

– Autres vecteurs: permet de choisir 5 têtes de vecteur différentes. En cliquant dans ce bouton une fenêtre de choix apparaît où l'on peut cliquer pour indiquer la taille choisie. Par défaut la taille est `Medium`.

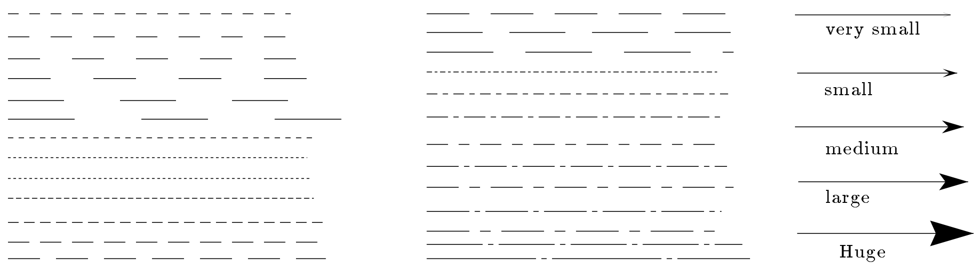


FIG. 2.12 - *Les traits pointillés et les vecteurs de* `xjpdraw`

Pour les traits épais le tracé à l'écran est relativement approximatif et il n'est donc pas souhaitable d'utiliser le bord des traits comme référence pour un dessin. Le style de trait courant est affiché dans une petite fenêtre de message placée dans le tableau des icônes. Il est aussi possible d'obtenir le style d'une primitive en activant les règles (voir le menu `PAGE`) et en sélectionnant la primitive: des informations sur cette primitive seront affichées dans la fenêtre "fast info".

2.11 Modifier un style de trait

On peut modifier à postériori le style des lignes, celui des primitives associées, celui de la primitive sélectionnée ou de la dernière primitive entrée. Pour cela il faut choisir le nouveau style dans le bouton des styles puis cliquer dans le bouton `Existant` (en-dessous du bouton des styles). Une fenêtre de choix apparaît avec un menu contenant `pour tous`, `pour les associées`, `pour la primitive sélectionnée`, `pour la dernière primitive entrée`, `pour les pointillés` ou `pour les traits épais`. La commande `Régler l'épaisseur des traits` permet de régler l'épaisseur des traits épais à l'aide d'un curseur. Elle peut être ajustée en 1/10 de mm jusqu'à 3mm. La valeur courante de l'épaisseur des traits épais est indiquée. La commande `pour les traits épais` permet de modifier à postériori l'épaisseur des traits des primitives tracées en trait épais. La commande `pour les pointillés` fait de même pour les traits pointillés.

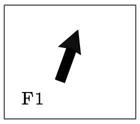
Noter que si le nouveau style fait intervenir à la fois des vecteurs et un autre changement de style il pourra être nécessaire de faire agir plusieurs fois le bouton de changement: par exemple un vecteur pointillé que l'on veut transformer en ligne fine sera d'abord transformé en ligne pointillé et il faudra cliquer à nouveau pour l'obtenir en ligne fine.

Chapitre 3

Sélection et association

La sélection d'une primitive est l'opération qui permet d'indiquer que l'on va effectuer une opération sur cette primitive. L'association consiste en fait à sélectionner plusieurs primitives simultanément.

3.1 Icône flêche en haut



En cliquant dans cette icône on va passer en mode "Sélection-Association". Dans ce mode il est aussi possible d'effectuer des commandes directes sans passer par les menus (voir page 102).

3.1.1 Sélectionner une primitive

Pour sélectionner placer la souris près de la frontière de la primitive choisie puis cliquer avec le bouton de gauche. Si la sélection est correcte la primitive clignote pendant un bref instant et la fenêtre "fast info" affiche le message "Primitive selected". Notez que vous pouvez modifier la fréquence de clignotement à l'aide de la commande `BLINK` de votre configurateur.

Ce mode est en fait géré par la commande `SELECTION_TYPE` du configurateur. Par défaut une primitive sélectionnée le reste jusqu'à la sélection suivante. Mais l'on peut définir aussi un mode où la dernière primitive crée devient automatiquement la primitive sélectionnée.

Si les règles (voir page 49) sont affichées les dimensions et caractéristiques de la primitive sélectionnée s'affiche dans la fenêtre "fast info". Noter que vous pouvez aussi obtenir des informations de distance et d'angle à l'aide de la boîte à mesure du menu `PAGE`.

Pour les primitives de type ligne, cercle, ellipse, rectangle, ovale après sélection apparaît une croix au centre de la primitive, pour les polygones une croix au barycentre.

A noter que pour accélérer la sélection le canevas de `xjpdraw` est découpé en petits carrés (les "buckets") et une liste associée à chaque primitive indique dans quels carrés on peut trouver la primitive. Si un problème apparaît lors de la sélection il est possible que les listes soient dans un état incorrect. La commande `Redessiner tout` du menu `PAGE` permet de remettre à jour les listes.

3.1.2 Sélection pour primitives recouvrantes

Si deux primitives p1, p2 se recouvrent il est parfois difficile de sélectionner une des deux (disons que p1 est toujours sélectionnée et que p2 ne l'est jamais). Dans ce cas il suffit de maintenir le bouton L3 appuyé lors de la sélection pour que la sélection s'opére sur p2. Le critère de choix pour la sélection se fait sur la distance puis sur l'ordre de création. L'appui sur L3 permet le choix de la primitive deuxième dans l'ordre des distances ou, pour des distances égales, deuxième dans l'ordre de création (sous réserve que la distance ne soit pas trop grande c'est-à-dire environ 1.5mm).

Il est aussi possible de passer dans un mode de sélection où seul un type donné de primitive peut être sélectionné: c'est le mode `sélection sélective`, voir page 45.

A noter que dans un certain sous-mode du mode "Sélection" (voir la section "Translater") une primitive sélectionné peut être translatée en maintenant le bouton de la souris appuyé et en la déplacant. Dans ce sous-mode il est possible qu'une primitive sélectionnée subisse un léger changement de position lors de la sélection.

3.2 Association

La notion d'association consiste à sélectionner plusieurs primitives simultanément pour leur faire subir un même traitement par la suite (par exemple translation pour les dessins, changement de style pour les textes...) Les primitives associées comportent sur leur frontière un signe "+". L'association peut s'effectuer de plusieurs manières.

- **Dans une primitive rectangle ou une primitive polygone:** On peut sélectionner toutes les primitives qui sont incluses dans une primitive de type rectangle ou un polygone en sélectionnant celle-ci. Pour cela il suffit d'appuyer sur la touche L8 après avoir sélectionné le rectangle ou le polygone. Un '+' apparaît dans un coin du rectangle ou du polygone, ainsi que dans toutes les primitives associées. On peut alors couper, copier, translater etc.. toutes les primitives en bloc selon les mêmes procédés que pour une primitive unique. Attention, une erreur fréquente s'obverve lors de la translation de primitives associées. Par inattention on peut amener des primitives en dehors de l'écran. Elles ne sont alors plus visibles à l'écran mais sont présentes sur le papier. Pour les supprimer il suffit de tout dissocier, de placer les primitives visibles dans un rectangle pour les associer puis de détruire les primitives non associées à l'aide de l'instruction `Détruire tout sauf les Associées` du menu `EDITION`. On peut aussi ramener les primitives hors-cadre dans le canevas à l'aide de la commande `Ramener une primitive hors cadre` du menu `PAGE`.

- **Dans un rectangle ou un cercle**: ceci sans que la primitive existe. En mode "Sélection" cliquer avec le bouton de gauche de la souris suffisamment loin de toute primitive et déplacer la souris en maintenant cliqué. Si vous utilisez le bouton de gauche un rectangle en pointillé va apparaître, si vous utilisez le bouton de droite ce sera un cercle. Toutes les primitives qui seront à l'intérieur de l'élément au moment où vous relacherez le bouton de la souris seront associées.

- **Primitives désignées:**

  On peut associer directement une primitive en restant dans le mode sélection mais en la sélectionnant avec le bouton du milieu.

- **Primitives du même type:** La commande `Association Multiple` du menu `EDITION` permet d'associer ou de dissocier sélectivement des primitives du même type (exemple: tous les textes). Le bouton de choix qui apparaît permet de sélectionner quel type de primitive on va associer ou dissocier.

- **Les primitives dans une certaine position par rapport à une autre**: la commande `Associer tout à ... de la sélectionnée` du menu `EDITION` permet d'associer toutes les primitives dans une position particulière par rapport à la primitive sélectionnée. On peut ainsi associer toutes les primitives à gauche, à droite, en dessous, au dessus de la primitive sélectionnée. On peut faire la même chose avec la primitive à droite et en dessous, à droite et au dessus, à gauche et en dessus ou à gauche et en dessous.

Pour dissocier (associer) toutes les primitives associées il suffit de choisir `Dissocier Tout` (`Associer tout`) dans le menu `EDITION`. En mode "Sélection" on peut le faire en tapant "D" ("A") dans le canevas. Dans les autres modes on peut utiliser `CTRL-a` et `CTRL-d`.

Pour dissocier une primitive il faut la sélectionner et choisir `Dissocier la Sélectionnée` dans le menu `EDITION`. En mode "Sélection" on peut le faire en tapant "d" dans le canevas après sélection de la primitive.

Les associées subiront en même temps les opérations suivante:

- translation
- miroir
- zoom
- texture

- destruction
- pivot
- alignement
- copie
- rotation

Chapitre 4

Manipulation de primitives

On va décrire ici quelques manipulations de base sur les primitives. On consultera le chapitre Menus pour une description détaillée des commandes.

4.1 Translater une primitive

Des translations peuvent être effectuées sur une primitive sélectionnée ou sur les primitives associées.

4.1.1 Translation à la souris

Après avoir sélectionné une primitive se rapprocher de cette primitive avec la souris, cliquer avec le bouton de gauche en maintenant cliqué, attendre la fin du clignotement puis déplacer la souris (lentement), toujours en maintenant appuyé.

La primitive suit le déplacement de la souris (figure 4.1a). Attention, ceci ne fonctionnera que si l'icône "Sélection" indique "`with mouse`" (sinon elle indique "`no mouse`"). Si cela n'est pas le cas appuyer préalablement sur la touche L6. Si le message n'apparaît pas vérifier dans le configurateur l'affectation de la touche L6 ainsi que la commande `FOLLOW_MOUSE`, voir le chapitre "Installation". Si les règles sont affichées la fenêtre fast info vous donnera la position de la primitive sélectionnée.

Remarque: Si l'on procède de cette manière pour translater un grand nombre de primitives associées le système va stocker toutes les positions intermédiaires (à une fréquence d'environ 1/40 de seconde) et retracer toutes les primitives à ces positions intermédiaires. Il est donc plus raisonnable dans ce cas d'opter pour la méthode des grands déplacements.

4.1.2 Translations importantes

Dans ce mode on indique tout d'abord un point sur la primitive puis sa position dans le futur. Il faut donc sélectionner un point de la primitive puis relacher la souris. Appuyer sur la touche L9 (une fenêtre indiquant que l'on a changé de mode doit apparaître) puis déplacer la souris à l'endroit où l'on veut placer le point sélectionné. Appuyer alors sur un bouton de la souris (figure 4.1b). Ce mode est persistent, donc n'oubliez pas d'en sortir en cliquant à nouveau sur L9. Noter que le programme vérifie si la translation ne met pas les primitives en dehors du cadre.

4.1.3 Translation fine

Vous pouvez aussi déplacer finement (de quelques pixels) une primitive. Il suffit d'avoir sélectionné la primitive puis d'utiliser les touches R1, R2 qui déplacent la primitive d'un pixel (environ 0.145mm) respectivement à gauche et à droite ou R4, R5 qui font la même chose vers le haut et le bas de la page.

4.1.4 Translation précise

Si l'on sait exactement de combien l'on doit translater une primitive (en cm) il est alors possible d'utiliser la commande `Translation précise` du menu `DISPOSITION`. Le programme attend alors dans la fenêtre de texte deux

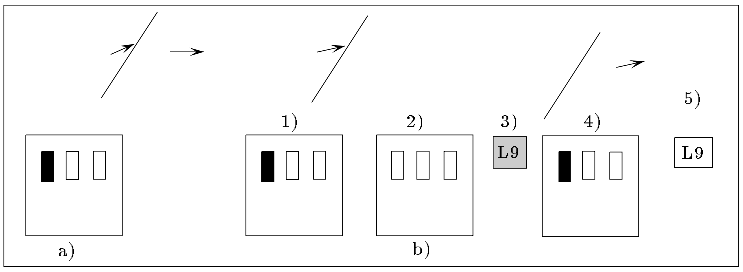


FIG. 4.1 - *Comment translater une primitive, a) petit déplacement, b) grand déplacement*

nombres qui sont les valeurs de la translation. Le programme vérifie que la translation indiquée ne met pas de primitives en dehors du cadre.
Attention: Pour une primitive texturée avec une texture de ligne (nous verrons plus loin comment cette opération est possible) la texture n'est pas associée physiquement à la primitive qui l'englobe. Donc si on la déplace, seul le contour extérieur se translate. Pour la translater en bloc il faut donc associer les lignes de textures à la primitive.

4.2 Modifier la taille d'une primitive

On peut modifier la taille de toutes les primitives de dessin. Pour cela il faut tout d'abord sélectionner une primitive, puis appuyer sur la touche L10. Une fenêtre indique que vous êtes dans ce mode. Ce mode est persistent, donc n'oubliez pas d'en sortir en retapant sur la touche L10. Si les règles sont affichées la fenêtre fast info vous donnera les dimensions de la primitive sélectionnée. Dans ce mode il n'est possible que d'éditer la primitive sélectionnée: pour en éditer une il faut sortir du mode, sélectionner et y revenir. La manière dont on édite dépend du type de la primitive, c'est ce qui va être décrit dans les sections suivantes.

4.2.1 Cas des rectangles

Mettez la souris près d'un des 4 coins et appuyer sur le bouton de gauche. La primitive change de forme jusqu'a ce que l'on relâche le bouton.

4.2.2 Cas des lignes

Mettez la souris près d'une des 2 extrémités et appuyer sur le bouton de gauche. Pour les lignes on peut ainsi faire tourner le segment autour de son autre extrémité.

Si l'on veut modifier la longueur d'une ligne sans en changer l'orientation il suffit d'appuyer une fois sur la touche `Shift` préalablement à la modification (une fenêtre indiquant que l'on est dans ce mode va apparaître et disparaîtra dès que l'on relachera le bouton de la souris). Pour changer l'orientation d'une ligne sans en changer la longueur utilisez le mode rotation. Si vous appuyer sur la touche L8 le changement de taille de la ligne se fera de façon à ce que sa pente soit compatible avec les valeurs des pentes de LaTeX standard.

4.2.3 Cas des cercles, ellipses, ovales

Pour une ellipse ou un ovale on peut modifier la longueur du petit ou du grand axe de la primitive. en cliquant avec le bouton de gauche près de la croix qui apparaît et en déplacant la souris en maintenant appuyé. Pour les cercles on peut choisir n'importe quel point sur le périmètre.

4.2.4 Cas des textes

Pour les primitives de texte la modification est d'un autre type: elle permet d'éditer le texte dans la mini-fenêtre `emacs` ou l'ensemble des textes dans une fenêtre qui va apparaître, (voir page 32).

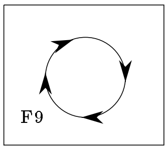
4.2.5 Cas des polygones, B-splines

Pour les polygones on peut soit modifier la position d'une arête existante ou en créer une nouvelle. Si le point cliqué est proche d'un sommet, il se place à la position courante de la souris. Si le point cliqué n'est pas un sommet on rajoute alors le point courant comme sommet du polygone (si c'est possible en fonction du nombre total de sommets du polygone, limité à 30). Ce comportement est valable aussi pour les B-splines et les courbes polynomiales, mais pas pour les splines.

4.2.6 Cas des splines

L'édition des splines se fait en déplacant les points de contrôle. Il suffit de cliquer avec le bouton de gauche a proximité d'un de ces points, puis de déplacer la souris en maintenant le bouton appuyé. Si l'on est trop loin d'un point de contrôle la position courante de la souris est ajoutée comme point de contrôle.

4.3 Rotation



Ce mode permet de faire tourner une primitive soit autour de son centre, soit autour d'un point quelconque. Les primitives associées tournent ensemble. Pour cela il suffit de sélectionner une primitive puis de cliquer sur l'icône de rotation (F9).

La primitive tourne autour de son centre (son barycentre pour un polygone). Si on désire une rotation autour d'un autre point il suffit d'appuyer sur le bouton L9 après avoir sélectionné l'icône de rotation. On indique alors le centre de rotation en cliquant avec le bouton de gauche de la souris. Le centre de rotation est indiqué par le caractère "+".

Pour indiquer l'angle de rotation on utilise le curseur qui apparaît dans la fenêtre des rotations (celle qui apparaît après avoir cliqué dans l'icône). Indiquer l'angle de rotation puis valider la rotation en appuyant sur le bouton `Oui`. Le bouton `Fin` permet de faire disparaître la fenêtre des rotations.

Une autre possibilité est de presser sur le bouton du milieu de la souris, puis de déplacer la souris en maintenant le bouton appuyé. L'angle de rotation (qui est affiché dans la fenêtre de rotation) est alors égal à l'angle entre:

– la droite passant par le centre de rotation et par le premier point cliqué

– la droite passant par le centre de rotation et par le point courant

Pour indiquer l'angle de rotation on peut aussi donner la valeur de l'angle en degré dans la fenêtre de texte qui apparaît dans la fenêtre de rotation.

Il est aussi possible de faire tourner les textes, mais pour que cela soit effectif sur le papier on doit utiliser une commande spéciale de LaTeX. Il est donc nécessaire d'inclure un fichier d'extensions LaTeX (usuellement `lplainext_dvips.tex`) dans votre source ou un package particulier si vous utilisez LaTeX2e (voir le chapitre "Installation").

4.3.1 Redémarrer

La commande `Redémarrer` permet la destruction complète de toute les primitives et la remise à l'état initial de `xjpdraw`. C'est la commande à utiliser pour commencer un nouveau dessin. Il est cependant plus prudent en général de sortir de `xjpdraw` et de le relancer.

4.4 Détruire une primitive

4.4.1 Détruire une primitive ou un ensemble de primitives

Après sélection (ou association) de la (des) primitive (s) à détruire on se rend dans le menu `EDITION` et l'on choisit la commande `Détruire la primitive sélectionnée` ou `Détruire les associées`. On peut aussi effectuer plus rapidement cette manipulation en conservant la souris dans le canevas et en tapant sur la touche fonction R13.

4.4.2 Où va la primitive détruite?

La primitive détruite reste en fait présente dans les structures de données: seule sa visibilité change. On peut faire apparaître à nouveau toutes les primitives détruites lors d'une session en utilisant la commande `Afficher le Presse-Papier` dans le menu `EDITION` (le raccourci est 'a' en mode "Sélection" et `CTRL-y` dans les autres modes).

On peut alors sélectionner une primitive du presse-papier et l'afficher à nouveau dans le dessin courant par la commande `Coller` du menu `Presse-Papier` dans le menu `EDITION` (raccourci: R7 en mode "Sélection"). La primitive disparaît du presse-papier. Si cette primitive est associée toutes les primitives associées sont à nouveau visible dans le dessin courant.

Les primitives détruites sont enregistrées dans le fichier `.macdraw` (sauf si l'on a vidé le presse-papier), on peut donc les récupérer à tout moment.

4.4.3 Détruire tout sauf ...

La commande `Détruire tout sauf les Associées` dans le menu `EDITION` permet de détruire toutes les primitives qui ne sont pas associées. La commande `Détruire tout sauf la Sélectionnée` dans le menu `EDITION` permet de détruire toutes les primitives sauf la primitive sélectionnée.

4.5 Correction de texte

Pour effectuer une correction de texte il existe une méthode simple utilisant un éditeur de type emacs. Pour faire apparaître dans la mini-fenêtre emacs un texte qui n'est pas le texte courant il suffit de le sélectionner puis d'enfoncer la touche L10 (qui permet en général un changement de forme d'une primitive). Le texte est alors présent dans la fenêtre `emacs` et peut être corrigé en positionnant le curseur avec les commandes `emacs` classiques:

- CTRL-a : curseur en début de texte
- CTRL-b : le curseur recule d'un caractère
- CTRL-d : on détruit le caractère précédant le curseur
- CTRL-e : curseur en fin de ligne
- CTRL-f : le curseur avance d'un caractère
- CTRL-t : on inverse les deux caractères précédents

puis en insérant ou en détruisant des caractères. On peut aussi utiliser la souris pour placer le curseur à l'endroit voulu. Il faut remarquer qu'un texte sur plusieurs lignes (donc comportant des retour chariot) sera affiché sur une

emacs:	■qwerty	CTRLa
emacs:	qwerty■	CTRLe
emacs:	qwert■y	CTRLb
emacs:	qwerty■	CTRLf
emacs:	qwert■	CTRLd
emacs:	qwetr■	CTRLt

FIG. 4.2 - *L'action des diverses commandes emacs sur un texte*

seule ligne, les retour chariot étant indiqués par un & (noter que si le & est précédé d'un anti-slash il n'est alors plus considéré comme un séparateur de ligne). On peut rajouter des lignes au texte en tapant ce caractère dans le texte mais il ne faut pas en placer deux consécutivement.

Pour les textes longs on peut utiliser le curseur au dessus du bouton `emacs` pour afficher la partie à corriger.

On peut donc rajouter des caractères, en détruire de manière commode. Il suffit alors de cliquer sur le bouton `emacs` pour valider les changements.

Si l'on veut faire un ensemble de corrections sur plusieurs textes on aura remarqué qu'en mode sélection, après avoir sélectionné un texte et appuyé sur la touche L10, une fenêtre apparaît contenant l'ensemble des textes visibles du dessin. On peut alors procéder dans cette fenêtre à des corrections sur tous les textes, là aussi en utilisant les commandes emacs (comme `CTRL-n, CTRL-p` pour aller à la ligne suivante ou précédente) ou la souris pour se positionner au bon endroit dans le texte à corriger. Ces corrections seront validées après avoir appuyé sur le bouton `Fin`. Si l'on ne désire pas valider les corrections il suffit de cliquer dans le bouton "Détruire la fenêtre". Noter que chaque texte qui apparaît dans cette fenêtre (par ailleurs munie de glisseur) est suivi de la chaîne `%at` suivie de la position (donnée en centimètre) du texte dans le dessin. Ceci permet de différentier des primitives ayant le même texte. De plus la primitive sélectionnée est indiquée par un `(X)` après le %.

4.6 Copier une primitive

Pour copier une primitive sélectionnée il faut choisir la commande `Copier` dans le menu `EDITION`. Un raccourci après la sélection est d'appuyer sur R6. La primitive copiée est strictement identique à la primitive sélectionnée (il y a donc recouvrement).

4.7 Copier une primitive avec homothétie

La commande `Copier avec homothétie` du menu `EDITION` permet de copier la primitive sélectionnée en exercant un facteur d'homothétie. Après invocation de cette commande une fenêtre apparaît avec un curseur qui permet d'ajuster le facteur d'homothétie. A chaque changement de ce facteur la primitive change de taille. Une fois la bonne taille obtenue il suffit de cliquer dans le bouton `Fin` pour valider la primitive. `Annuler` permet d'annuler cette commande. L'homothétie s'exerce de la manière suivante selon le type de la primitive:

- texte: pas de changement
- ligne: l'origine du segment est conservée et la longueur est multipliée par le facteur d'homothétie.
- rectangle, cercle, ellipse, ovale: le centre de ces primitives reste invariant et les dimensions sont multipliées par le facteur d'homothétie.
- polygone: le barycentre du polygone reste invariant.

Si cette commande agit sur des primitives associées seule la primitive sélectionnée subit l'homothétie.

4.8 Les miroirs, pivot, symétrie et le zoom

Ces trois commandes sont décrites précisément dans le chapitre consacré aux menus (menu `DISPOSITION` et menu `PAGE`).

4.9 Couper une primitive

Pour créer une coupure sur une primitive existante il suffit de choisir dans le menu des icônes l'utilitaires `ciseaux`. On doit tout d'abord indiquer la primitive que l'on va couper (en cliquant avec le bouton de gauche près de sa frontière) puis indiquer le point de coupure avec le bouton par un double clique sur le bouton du milieu. La primitive est transformée en ligne polygonale. Avant chaque coupure il est nécessaire de procéder à une sélection.

4.10 Loupe

Cet utilitaire permet d'obtenir un grossissement d'un facteur 3 sur une partie du dessin. Pour cela choisir dans le menu des utilitaires la loupe. Une fenêtre carré apparaît dans le canevas et l'on y affiche la version grossie du dessin dans la zone où se trouve la souris. Il est possible dans ce mode d'utiliser les touches R1, R2, R4, R5 pour translater les primitives d'un pixel si une primitive a été sélectionnée avant: la fenêtre affiche alors le nouveau résultat. Pour faire disparaître la loupe il suffit de cliquer dans une icône.

4.11 L'Aimant

L'aimant[1] est un mode qui permet de simuler une modification artificielle de la position de la souris pour agir comme si elle était placée en un autre point de l'écran. Cette modification est gérée par un mode qui est choisie dans la fenêtre à droite de l'écran. On trouve dans cette fenêtre:

- `le + proche` : on agit comme si la position de la souris était celle du point le proche sur la primitive la plus proche
- `// +proche` : valable seulement pour la création de droite et de polygone. Le point final de la ligne ou d'une arête sera modifié de façon à ce que la ligne ou l'arête soit parallèle à la face la plus proche de la primitive la plus proche. La détermination de la primitive la plus proche se fait par rapport au point de départ de la ligne ou de l'arête. On regarde le point A le plus proche sur la primitive la plus proche de ce point, on détermine la pente de l'arête de la primitive la plus proche au point A, qui sera la pente de la nouvelle primitive.
- `sommet`: le point modifié sera le sommet de la primitive la plus proche. Pour les segments cela signifie l'extrémité, pour les cercles, ellipses et ovales le centre, pour les rectangles soit un sommet soit le centre.
- `inter`: si l'on se trouve très près d'un point d'intersection de deux primitives le point modifié sera alors le point intersection
- `Inactif`: le mode normal, la position de la souris est la position courante.

Cette fonction est particulièrement utile lors de la création des primitives (figure 4.3). A noter que si l'on est dans un mode actif une primitive peut ne pas apparaître en cours de création si `xjpdraw` ne trouve pas de position satisfaisant le critère choisi, il faut alors modifier la position de la souris jusqu'à ce qu'une primitive apparaisse.

Noter que l'on peut changer de mode à tout moment en utilisant les séquences `CTRL-n` (mode suivant) et `CTRL-p` (mode précédent), de manière cyclique. C'est en utilisant ce type de gymnastique que l'on peut crée une ligne partant d'un point d'intersection et terminant au sommet d'un carré : on active tout d'abord l'aimant en `Inter` puis en maintenant le bouton pressé on utilise `CTRL-p` pour passer en mode `sommet`. La fenêtre de l'aimant peut être fermée en cliquant dans le bouton `Aimant` de la fenêtre des utilitaires. Elle peut être réaffichée de la même manière.

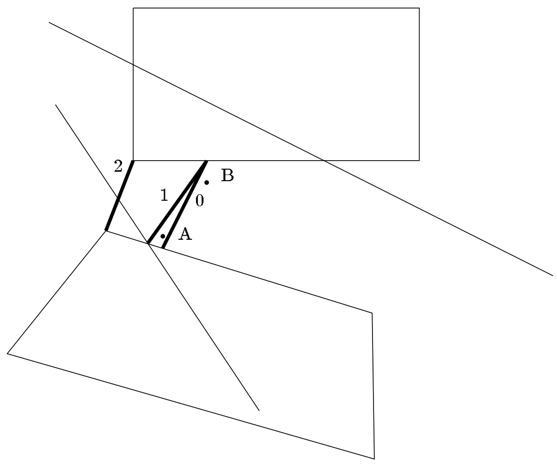


FIG. 4.3 - *Action de l'aimant pour la création d'une ligne. On a cliqué au point A, on s'est déplacé et on a relaché le bouton en B. Pour la ligne 0 on était dans le mode* `+ proche`, *pour la ligne 1 on a commencé en mode* `Inter` *et en cours de création on est passé au mode* `+ proche`. *Pour la ligne 2 on était en mode* `sommet`.

1. L'idée de l'aimant est du à O. Schwartzkopf qui l'a implanté dans son système IPE. Elle a été simplement reprise dans `xjpdraw`, en y ajoutant un mode

Chapitre 5

xjpdraw et Postscript

5.1 Pourquoi Postscript?

Nous avons vu que `xjpdraw` peut fonctionner du point de vue graphique sous deux modes. Le premier utilise les primitives graphiques standardes de LaTeX, le second est celui où le texte est sauvé en LaTeX et les primitives graphiques au format Postscript. C'est cette méthode qui est recommandée et que nous allons examiner. Le choix fait dans `xjpdraw` de conserver la définition des textes à l'aide d'ordre LaTeX alors que l'on aurait pu tout écrire en Postscript se justifie par le fait que l'on conserve pour les textes des dessins la cohérence avec la typographie du texte source (ce qui ne serait pas le cas en Postscript) et que l'on peut aussi utiliser toute la puissance de traitement de LaTeX pour les textes inclus dans un dessin.

Il se pose donc la question de savoir comment on peut mélanger des ordres LaTeX et des dessins décrit en Postscript.

5.2 Comment ca marche?

Donnons en un exemple. On va générer un dessin simple avec du texte (le mot `toto`) et une ligne et on va le sauver sous le mode décrit et regardons ce qui se passe au niveau des fichiers crées. On voit que `xjpdraw` génére alors trois fichiers `xx.ltex`, `xx.macdraw`, et `xx.ps` (ou `xx.ps.Z`).
Examinons tout d'abord le fichier LaTeX. Il se présente sous la forme suivante:

```
% dessin effectue par JpDraw (un programme de JP. MERLET)
% INRIA Sophia-Antipolis,France le: Thu Jun  8 00:18:54 1995
% Directory de creation:/u/merlet
% Createur:merlet
\ifx\cs\tempdima \newdimen \tempdima\fi
\ifx\cs\tempdimb \newdimen \tempdimb\fi
\newcount \jpdrawx
\jpdrawx=100 \tempdima=   1.00cm \tempdimb= 1.0cm
\ifdim \unitlength = 1pt \unitlength=  1.00cm \fi
\ifdim \unitlength > \tempdima \tempdima=\unitlength \fi
\ifdim \unitlength < \tempdima \tempdima=\unitlength \fi
\multiply \jpdrawx by \tempdima \divide \jpdrawx by \tempdimb
\setlength{\unitlength}{\tempdima}

\begin{picture}(   2.0,   2.4)(   1.3,   9.9)
\thinlines
\large
\put( 1.334, 9.889){\special{psfile="`zcat /u/merlet/xx.ps" hscale=\the\jpdrawx \sp vscale=\the\jpdrawx}}
\scriptsize
\put(2.52,10.90){toto}
\end{picture}
\setlength{\unitlength}{1cm}
```

Dans ce fichier les quatre première lignes résument l'historique de la création du dessin. Les lignes suivantes (jusqu'au début de l'environnement `picture`) permettent d'affecter à un compteur `jpdrawx` une valeur qui vaut par défaut 100.

Si la variable `unitlength` est définie à une valeur différente de 1 cm alors la valeur du compteur `jpdrawx` est calculée comme la partie entière de `unitlength` multipliée par 100. Ainsi une valeur de 0.8cm conduit à une valeur de `jpdrawx` de 80. Rappelons que `unitlength` est une dimension au sens TeX qui permet d'affecter une mesure à l'unité utilisée dans l'environnement `picture`. Ainsi si on affecte la valeur 1cm à `unitlength` notre dessin occupera un rectangle de 2cm par 2.4cm.

L'ordre d'insertion du fichier Postscript apparaît juste après le début de l'environnement `picture`

```
\put( 1.334, 9.889){\special{psfile="`zcat /u/merlet/xx.ps"...
```

et utilise la commande `special` de LaTeX. Dans la suite nous supposerons que le programme utilisé pour convertir les fichiers .dvi en postscript est `dvips`. Une certaine latitude est cependant laissé à ce sujet, en particulier en modifiant les variables `DVI` et `PS` du configurateur, voir le chapitre "Installation".

Ce driver comprend le mot-clé `psfile` qui permet d'inclure un fichier Postscript. Ici on a utilisé une astuce de ce driver en utilisant le mot `zcat` qui indique que l'on va inclure non pas le fichier `xx.ps` mais le fichier `xx.ps.Z` compressé par l'utilitaire `compress`.

Dans la suite de l'ordre d'inclusion on remarque que les mot-clés `hscale, vscale` sont affecté à la valeur du compteur `jpdrawx`. Ces mot-clés permettent d'effectuer une mise à l'échelle du Postscript inclus: une valeur de 100 ne change pas les dimensions alors qu'un valeur inférieure va le réduire. Comme nous avons vu que ce compteur était directement calculé à partir de la valeur de `unitlength` on se rend ainsi compte que modifier la valeur de `unitlength` permet de réduire ou d'augmenter à volonté la taille du dessin. Attention cependant au grossissement car l'épaisseur des lignes épaisses est elle aussi augmentée.

Le dernier élément de l'environnement `picture` est celui qui permet d'écrire le texte `toto` à une position voulue. Comme la position est définie en unité arbitraire la modification de la variable `unitlength` modifiera sa position. Comme le Postscript est aussi modifié le changement de taille est donc valide aussi bien pour le dessin que pour le texte.

Le fichier Postscript `xx.ps` se présente sous la forme suivante:

```
%%BoundingBox: 0 0 56 68
% initial value of unitlength : 1.000000
/Ellipse { gsave moveto setgray /savematrix matrix
currentmatrix def currentpoint translate rotate newpath
scale arc closepath savematrix setmatrix stroke grestore 0 setgray} def
/EllipseColor { gsave moveto setgray /savematrix matrix
currentmatrix def currentpoint translate rotate newpath
scale arc closepath savematrix setmatrix fill grestore 0 setgray} def
/EllipseRVB { gsave moveto setrgbcolor /savematrix matrix
currentmatrix def currentpoint translate rotate newpath
scale arc closepath savematrix setmatrix fill grestore 0 setgray} def
/EllipseBRVB { gsave moveto setrgbcolor /savematrix matrix
currentmatrix def currentpoint translate rotate newpath
scale arc closepath savematrix setmatrix stroke grestore 0 setgray} def
/Ligne { gsave moveto currentpoint translate 0 0 moveto
lineto currentpoint stroke moveto closepath grestore} def
/LigneRVB { gsave setrgbcolor moveto currentpoint translate 0 0 moveto
lineto currentpoint stroke moveto closepath grestore 0 setgray} def
/ArrowNoire {gsave 0 setgray moveto currentpoint
translate 0 0 moveto lineto lineto lineto lineto
currentpoint fill moveto closepath grestore 0 setgray} def
/ArrowRVB {gsave setrgbcolor moveto currentpoint
translate 0 0 moveto lineto lineto lineto lineto
currentpoint fill moveto closepath grestore 0 setgray} def
/RectangleNoir {gsave setgray moveto currentpoint
translate 0 0 moveto lineto lineto lineto lineto
currentpoint fill moveto closepath grestore 0 setgray} def
/RectangleRVB {gsave setrgbcolor moveto currentpoint
translate 0 0 moveto lineto lineto lineto lineto
currentpoint fill moveto closepath grestore 0 setgray} def
/Rectangle {gsave moveto currentpoint
translate 0 0 moveto lineto lineto lineto lineto
currentpoint stroke moveto closepath grestore} def
```

```
/OpenPoly {gsave moveto currentpoint translate 0 0 moveto} def
/Li {lineto} def
/ClosePnoir {currentpoint fill moveto closepath grestore 0 setgray} def
/ClosePoly {currentpoint stroke moveto closepath grestore 0 setgray} def
gsave
 /ulx where{pop ulx uly scale} if
 %currentpoint translate
0.3 setlinewidth
[] 0  setdash
 55.899  67.819   0.000   0.000 Ligne grestore
```

La première partie, en commentaire, indique la BoundingBox du dessin et la valeur de la variable `unitlength` utilisée lors de la création du dessin. On trouve ensuite la définition de diverses procédures Postscript. Le tracé de ligne est effectué à l'avant-dernière ligne du fichier, en utilisant une de ces procédures:

```
 55.899  67.819   0.000   0.000 Ligne grestore
```

Pour l'inclusion du dessin les choses restent strictement transparente pour l'utilisateur. On utilise la procédure habituelle (un input du fichier `xx.ltex` dans le fichier LaTeX), ce fichier faisant directement appel au fichier `xx.ps`. Il conviendra toutefois d'être prudent si l'on déplace ces fichiers de les déplacer tous et de mettre à jour dans l'ordre d'inclusion du fichier Postscript le chemin permettant d'accéder à ce fichier. A partir du fichier source suivant:

```
\documentstyle{article}
\begin{document}
\begin{figure}[hbtp]
\begin{center}
\input{xx.ltex}
\caption{Le dessin de test}
\end{center}
\end{figure}
\end{document}
```

on obtient le dessin de la figure 5.1.

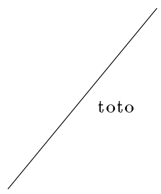


FIG. 5.1 - *Le dessin de test*

Une dernière astuce est possible avec `unitlength`. Si l'on affecte une valeur négative à cette dimension juste avant l'inclusion d'un dessin la dimension maximale du dessin sera égale à la valeur absolue de `unitlength`. Ainsi dans notre exemple affecter -4.8cm à `unitlength` ferait que l'on aurait un dessin de 4cm par 4.8cm.

5.3 La commande special

Cette manière de procéder est simple mais souffre d'un inconvénient. En effet les ordres d'inclusion des fichiers postscript différent selon le type de programme qui est utilisé pour la conversion du .dvi en fichier postscript. Dans l'exemple nous avons ainsi supposé que le programme utilisé est `dvips`. Toutefois si ce n'est pas la syntaxe utilisée par le programme que vous utilisez pour convertir un fichier `.dvi` en Postscript vous pouvez modifier cette instruction dans le configurateur (voir le chapitre "Installation").

Noter que si vous utilisez `dvips` et que dans le configurateur vous avez indiqué dans l'ordre d'inclusion le mot `zcat` alors `xjpdraw` va automatiquement compresser le fichier postscript à l'aide de la commande `compress`.

Un autre mode de fonctionnement est obtenu lorsque l'on sauve en utilisant l'instruction **`Sauver PS et LaTeX dans le fichier`**. Dans ce cas la totalité du postscript est inclus dans le fichier LaTeX. Bien que plus onéreuse en taille mémoire (le Postscript n'est pas compressé) c'est la bonne manière de faire si l'on veut créer un document contenant toute les informations pour l'impression, par exemple dans le cas de l'envoi d'un article par courrier électronique.

5.4 Textures et textes

Nous verrons par la suite qu'il est possible de texturer les primitives avec une texture à niveau de gris ou en couleur. Supposons maintenant que l'on place du texte à l'intérieur de primitives texturées. Il faut savoir qu'en Postscript une texture masque complètement toute primitive écrite précédemment à la même position (par exemple une texture blanche dans une ellipse masquera complètement tout autre élément ou partie d'élément contenue dans l'ellipse). Cette particularité peut compliquer l'interaction entre les textes et le dessin. Par exemple si un texte apparaît sur une primitive coloriée et que l'ordre d'inclusion du fichier Postscript est après l'ordre d'écriture du texte alors le texte n'apparaîtra pas sur le dessin. Par défaut dans `xjpdraw` l'ordre d'inclusion du Postscript est le premier mais vous pouvez modifier cet ordre en utilisant une commande du menu `FICHIER` (voir page 41).

5.5 Inclure un fichier Postscript

Il est possible d'insérer dans un fichier `xjpdraw` une figure Postscript qui a été généré par un autre programme (par exemple `Maple`). Pour cela `xjpdraw` ne cherche pas à interpréter ce dessin pour le reconstruire au sens des primitives. Il se borne à inclure dans le fichier LaTeX le postscript qui lui a été fourni. L'affichage repose sur l'utilisation du visualisateur de Postscript `gs` qui permet de produire un fichier bitmap du dessin. Ce bitmap est alors simplement reproduit dans le canevas `xjpdraw` à condition que l'opérateur fournisse les dimensions du dessin.

Il ne sera pas possible de manipuler ou d'éditer le dessin inclus, seul les translations et les zooms seront possibles, ainsi que, bien sûr le recouvrement par des primitives `xjpdraw`. Pour plus de détails voir page 42.

Chapitre 6

Description détaillée des menus

Dans ce chapitre nous allons décrire précisément les différentes commandes des menus. Les menus sont: `EXTENSION`, `FICHIER`, `EDITION`, `STYLE` , `CARAC`, `PAGE`, `DISPOSITION`, `Externe`, `ANNULER` et `BIB`

Ces menus ouvrent l'accès à différentes commandes qui apparaissent si l'on met la souris dans le cadre du menu et que l'on maintient cliqué le bouton de droite de la souris. Pour passer d'un menu à un autre il est nécessaire de relacher le bouton de la souris.

6.1 Le bouton `Extensions`

Le bouton `Extensions` permet soit de n'utiliser que les possibilité graphiques standardes de LaTeX ou au contraire d'utiliser pleinement le Postscript. Si on clique dans ce bouton le message passe à `Standard` et dans ce cas on ne peut générer que des fichiers compatibles avec le standard graphique de LaTeX, version 2.09. On se retrouve donc avec les contraintes correspondantes et `xjpdraw` gére ces contraintes. Ainsi les lignes sont tracées avec une pente compatible avec LaTeX car `xjpdraw` modifie l'extrémité de la ligne pour cela. La contrainte sur le diamètre maximum des cercles n'est cependant pas prise en compte. Pour sauver un dessin crée dans le mode `Standard` utilisez les ordres `Sauver` ou `Sauver sous` du menu `FICHIER`.

Si l'on est en mode `Standard` et que l'on tente d'effectuer une opération pour laquelle il n'existe pas d'ordre LaTeX une fenêtre apparaît. On y indique que l'opération n'est pas permise et on demande si l'on doit passer en mode étendu.

Attention dans le mode fonction utilisateur (voir le chapitre "Fonction Utilisateur") il n'y a pas de vérification de la compatibilité avec le LaTeX standard à la création mais uniquement à l'enregistrement. L'enregistrement sans les extensions ne peut se faire que si on est en version standard.

6.2 Le Menu `FICHIER`

6.2.1 Redémarrer

Cette commande fait un effacement de l'écran et remet `xjpdraw` dans un état permettant la création d'un nouveau fichier. Si le fichier courant n'a pas été sauvé une fenêtre apparaît associée à des commandes: `continuer`, `enregistrer`, `annuler`. La première exécute `Redémarrer` sans mise à jour du fichier courant, `enregistrer` sauve les modifications sur le fichier courant et `annuler` annule la commande `Redémarrer`.

6.2.2 Charger

Cette commande permet de charger un fichier. Les seuls fichiers que l'on peut recharger sont ceux sous format `macdraw`. `xjpdraw` lit aussi les fichiers `.macdraw` qui ont été compressés par `compress` (extension .Z) ou `gunzip` (extension .gz).

Une fenêtre apparaît, contenant différentes fenêtres décrivant les fichiers `macdraw` disponibles dans le répertoire courant, ainsi que les sous-répertoires présents (le répertoire supérieur est indiqué par `..`). On peut alors sélectionner un fichier à charger en cliquant dans son nom. Si l'on clique dans un nom de répertoire la fenêtre des fichiers disponibles est mise à jour.

On peut aussi indiquer le nom d'un fichier à charger dans la fenêtre de texte, en bas à droite. Pour cela on peut soit indiquer un nom complet en omettant l'extension `.macdraw` si la chaîne de caractères ne contient pas d'autre extension, en terminant par un `Return`. On peut aussi indiquer un début de nom et demander la complétion en terminant par un espace ou un caractère de tabulation. Dans ce cas la fenêtre des fichiers montrera tous les fichiers dont les noms commencent par la chaîne donnée. On peut alors cliquer dans un de ces fichiers pour le charger. Le caractère ~ placé en début de nom sera remplacé par le contenu de la variable d'environnement `HOME`.

A noter que la fenêtre des fichiers ne disparait que si l'on clique dans le bouton `Quit` de la fenêtre des fichiers.

On peut aussi rechercher un fichier dont on ne connait qu'imparfaitement le nom et la localisation: pour cela donner le nom dans la fenêtre `Recherche` de la fenêtre des fichiers en utilisant le caractère `*` pour indiquer un groupe de lettres inconnu (par exemple `toto*` permettra de retrouver le fichier `toto-nouveau`). Le fichier est recherché à partir du répertoire courant et dans tous les sous-répertoires. La complétion avec le caractère d'espace fonctionne aussi dans ce mode.

Pour faciliter la lecture dans la fenêtre des fichiers on peut demander en cliquant dans un bouton de ne pas afficher les fichiers de sauvegarde (qui commence par un #C), un autre bouton permettant même de détruire tous ces fichiers.

Attention, un fichier crée en mode transparent (voir le chapitre correspondant) ne pourra pas être chargé si `xjpdraw` n'est pas en mode transparent.

6.2.3 Sauver

Dans toute cette section lorsque l'on indique que l'on doit donner un nom de fichier il faut introduire ce nom dans la fenêtre de texte et terminer par un `Return`.

6.2.3.1 Sauver

Cette commande permet l'enregistrement du dessin dans le fichier courant au format graphique standard de LaTeX.

6.2.3.2 Sauver Sous

Cette commande permet de sauvegarder un fichier sous un autre nom que le nom courant, au même format que `Sauver`. Donner le nom du fichier (il ne doit pas avoir l'extension `.macdraw`). En cas d'erreur de manipulation il suffit de donner un nom vide pour annuler cette commande. Le caractère ~ placé en début de nom sera remplacé par le contenu de la variable d'environnemnt `HOME`.

6.2.3.3 Sauver sous format bitmap

Cette commande permet de sauver un dessin dans le format bitmap. Donner le nom du fichier. En cas d'erreur de manipulation il suffit de donner un nom vide pour annuler cette commande.

6.2.3.4 Sauver sous format inrimage

Attention cette fonctionnalité est actuellement inopérante.

6.2.3.5 Sauver le Dessin en Postscript et le Texte en LaTeX

Cette commande permet de sauver le dessin dans le fichier courant, en créant un fichier LaTeX qui inclut un fichier Postscript d'extension `.ps`, voir page 35. Un raccourci pour cette commande est `CTRL-w`. Pour connaître le nom du fichier courant utilisez `CTRL-f`.

6.2.3.6 Sauver le Dessin en Postscript et le Texte en LaTeX Sous

Idem que précédemment mais on doit d'abord indiquer le nom d'un fichier LaTeX. Donnez le nom du fichier. En cas d'erreur de manipulation il suffit de donner un nom vide pour annuler cette commande. Le caractère ~ placé en début de nom sera remplacé par le contenu de la variable d'environnement `HOME`.

6.2.3.7 Sauver PS et LaTeX dans le fichier

Cette commande permet de sauver le texte et le dessin au format Postscript dans le fichier LaTeX courant. A l'opposé de la commande précédente il n'y a pas de création d'un fichier Postscript, les commandes Postscript étant directement incluse dans le fichier LaTeX.

6.2.3.8 Sauver le Dessin en Postscript et le Texte en LaTeX Sous

Idem que précédemment mais on doit d'abord indiquer le nom d'un fichier LaTeX. Donnez le nom du fichier. En cas d'erreur de manipulation il suffit de donner un nom vide pour annuler cette commande. Le caractère ~ placé en début de nom sera remplacé par le contenu de la variable d'environnement `HOME`.

6.2.4 Impression Ecran

Attention cette fonctionnalité est actuellement inopérante.

6.2.5 Vérifier

On a le choix entre `Vérification LaTeX` et `Vérification LaTeX avec Postscript`. Dans le premier cas on a crée le dessin dans le mode `Standard` et dans le second les primitives graphiques sont en Postscript.

Cette commande crée un fichier `bidon.ltex` contenant le dessin courant et qui est directement compilable par LaTeX. Dans le second cas le fichier Postscript du dessin s'appelle `xxx.ps`. Cette fonctionnalité ne marchera que si `xjpdraw` a été correctement installé (voir le chapitre "Installation").

6.2.6 Album

Cette commande permet de rajouter au dessin courant un dessin défini dans un autre fichier `.macdraw`. La commande demande le nom du fichier `.macdraw` et le mode de chargement est identique que pour la commande `Charger`.

L'extension `.macdraw` n'est nécessaire que si une autre extension est présente dans le nom du fichier. Le caractère ~ placé en début de nom sera remplacé par le contenu de la variable d'environnement `HOME`.

Les primitives de l'album sont associées pour que l'on puisse les déplacer facilement: en conséquence une bonne pratique est de dissocier toutes les primitives avant d'utiliser cette commande. `xjpdraw` lit aussi les fichiers `.macdraw` qui ont été compressés par `compress` (extension .Z) ou `gunzip` (extension .gz).

6.2.7 Mettre le Postscript en fin du LaTeX

Cet ordre permet de placer l'instruction d'inclusion du fichier Postscript en dernière position dans le fichier LaTeX généré par `xjpdraw`. Cela aura pour conséquence que tout texte placé sur une primitive coloriée va disparaître.

6.2.8 Mettre le Postscript au début du LaTeX

Cet ordre permet de placer l'instruction d'inclusion du fichier Postscript en première position dans le fichier LaTeX généré par `xjpdraw`. Cela aura pour conséquence que tout texte placé sur une primitive coloriée va apparaître même si elle est cachée à l'écran. C'est l'option par défaut.

6.2.9 Lire un configurateur

Cette commande permet de lire un fichier configurateur (voir le chapitre "Installation"). Le nom du fichier est demandé dans la fenêtre de texte.

6.2.10 Compiler et imprimer

Cette commande permet de créer un fichier `bidon.ltex` qui contient uniquement le dessin et de l'imprimer. Pour que cette fonctionnalité soit opérante il est nécessaire que la commande `PRINTING` du configurateur ait été installée (voir le chapitre "Installation").

6.2.11 Compiler et visualiser

Cette commande permet de de compiler le dessin courant (ce qui est assuré depuis la version 4.2 par une tâche de fond pour en assurer la rapidité) et de le transformer en postscript, puis de le visualiser. Pour que cette fonctionnalité soit opérante il est nécessaire que la commande `GHOST` du configurateur ait été installée (voir le chapitre "Installation"). Le visualisateur `ghostview` paraît bien approprié à cette commande. Noter que le nom du fichier servant à générer le fichier Postscript s'appelle `bidon1.dvi`. Un raccourci pour cette commande est `CTRL-e`.

6.2.12 Recompiler pour le previewer

Cette commande permet de créer un fichier `bidon.ltex` qui contient simplement le dessin et de le compiler. Si préalablement vous avez utilisé la commande `Compiler et visualiser` pour visualiser votre dessin et que vous avez conservé le visualisateur Postscript vous pourrez voir la nouvelle version du dessin simplement en rechargeant le fichier Postscript. que le nom du fichier servant à générer le fichier Postscript s'appelle `bidon1.dvi` et que le nom du fichier Postscript généré doit être cohérent avec le nom donné dans la commande `GHOST`. Le raccourci pour cette commande est CTRL-x.

6.2.13 Inclure un fichier Postscript

Cette commande permet d'inclure dans un dessin `xjpdraw` un fichier postscript qui n'a pas été crée par `xjpdraw`. Le nom du fichier ainsi que sa hauteur et sa largeur (en centimètres) est demandé dans la fenêtre de texte. Pour utiliser cette fonctionnalité il est nécessaire d'avoir accès à la ressource `gs` et d'avoir placé cette ressource dans le fichier configurateur (via la commande `GS`, voir le chapitre "Installation").

Une image au format ppm est crée par `gs` et recopié sous forme de bitmap à l'écran avec le coin supérieur de l'image placé en (0,0) (on pourra la translater par la suite). En conséquence le chargement est un peu long (la fenêtre "fast info" indique où en est le processus). N'oubliez pas que ce qui est montré à l'écran est environ deux fois plus gros que la réalité du papier, ce qui explique la qualité moyenne de l'image.

L'image est entourée d'un rectangle, au sommet duquel on a la mention `Fichier Postscript`, c'est ce cadre que l'on doit sélectionner pour sélectionner ce type de primitive. Une fois que l'image est affichée à l'écran elle est recopiée en interne de façon à ce que le réaffichage de la primitive soit très rapide. Cette manière de faire à l'inconvénient que l'image sera partiellement incorrecte si la primitive dépasse la partie visible du canevas. Il est bien sûr aussi nécessaire que la taille du dessin n'excède pas la taille de la page contenu dans le canevas (`xjpdraw` le vérifie et refusera les dimensions incorrectes).

Avant d'inclure un fichier Postscript il est donc nécessaire d'estimer sa taille (c'est-à-dire les plus grandes largeur, hauteur entre les traits visibles). Cela peut se faire en examinant la `BoundingBox` du dessin, s'il y en a une. Les deux premiers chiffres indiquent la position du coin inférieure gauche du dessin, les deux suivants les coordonnées du coin supérieur droit. Noter que l'unité des nombres suivant la BoundingBox peut être le pt, c'est-à-dire 2.54/72cm. Il sera cependant parfois nécessaire d'imprimer le dessin pour calculer sa taille car certains programmes ne propose pas de BoundingBox ou en fournisse une incorrecte. Si votre dessin est trop grand il peut être nécessaire de modifier le fichier Postscript pour le réduire. Pour cela on utilise l'ordre Postscript `scale` précédé de deux nombres indiquant un facteur d'échelle en x et en y. Ainsi

```
0.5 0.5 scale
```

va réduire votre dessin de moitié. Pour déterminer où vous devez placer cet ordre sachez qu'un fichier Postscript est généralement constitué de deux parties: un dictionnaire définissant des procédures (dont les noms commencent par un `/`) et le dessin lui même. Il sera bon de placer l'ordre `scale` juste après le dictionnaire, devant le dessin.

Il a été aussi parfois observé un décalage entre le dessin apparaissant à l'écran et celui produit sur le papier: ce qui est génant car, si vous ajoutez des primitives dans une position particulière par rapport au Postscript inclus, elles ne seront pas dans une position correcte sur le papier. L'astuce pour corriger cela est faire son dessin complètement en positionnant les primitives `xjpdraw` par rapport à ce qui apparaît à l'écran, de sauver puis de corriger manuellement dans le fichier LaTeX la position où l'on demande d'insérer le Postscript externe de façon à ce que le dessin soit correctement imprimé. Par exemple si vous avez dans le fichier LaTeX la commande:

```
\put( 1,10){\special{psfile="/u/merlet/postscript_externe.ps"}}
```

il suffit de corriger les chiffres (1,10). Un autre problème est que parfois après avoir inclus un dessin contenant ce type de primitive dans votre document principal, il ne soit plus possible d'imprimer les pages suivantes. Il n'y a pas de remède miracle mais vous pouvez toujours supprimer la première ligne de votre fichier Postscript si elle commence par `%!` ainsi que la commande `showpage` qui peut apparaître vers la fin de votre document.

Un fichier postscript inclus de cette manière ne peut pas être édité mais seulement translaté. Il est toutefois possible de procéder à une réduction ou un grossissement en utilisant la méthode décrite page 45. On peut aussi recouvrir le dessin de primitives `xjpdraw` (voir le dessin de la page 122).

6.2.14 Voir tous les dessins d'un répertoire

Cette instruction permet de visualiser tous les dessins `xjpdraw` du répertoire courant. Ces dessins sont affichés par groupe de 4, une fenêtre permet de continuer la visualisation ou de l'arrêter. On voit aussi les dessins dont les `.macdraw` ont l'extension .Z et .gz, donc les fichiers compressés par `compress` ou `gunzip`.

6.2.15 Sauver, Lire le contexte

Un contexte décrit l'état actuel de `xjpdraw` du point de vue de l'environnement graphique (par exemple présence d'un quadrillage, de règles, épaisseur des traits, etc..). On peut sauver un contexte puis le relire pour se retrouver dans le même état graphique. A noter que l'on peut systématiquement travailler avec un contexte en le mettant dans le configurateur (voir le chapitre "Installation").

6.2.16 Mode transparent

Cet ordre permet de passer en mode transparent et de créer la première page. On préférera en général lancer `xjpdraw` avec l'option `-slide` qui aura le même effet.

6.2.17 Nouvelle page

Permet de créer une nouvelle page en mode transparent.

6.2.18 Aller à la page..

Permet de naviguer dans les pages d'un document lorsque l'on est en mode transparent.

6.2.19 Ajouter une page

Crée une page vide après la page courante.

6.2.20 Détruire la page

Détruit la page courante et renumérote les pages suivantes.

6.2.21 Quitter

Avec cette commande vous sortez de `xjpdraw`. On demandera confirmation de la sortie et on signale si le fichier a été modifié sans être sauvé.

6.3 Le Menu `EDITION`

Toute les commandes de ce menu concerne la transformation de primitives.

6.3.1 Détruire

Cette section concerne la destruction de primitives. Les primitives détruites sont mises dans le presse-papier et peuvent être récupéré, sauf si vous videz le presse-papier. Elles sont aussi sauvées dans le fichier `.macdraw` et peuvent donc être récupérées après sortie de `xjpdraw`.

6.3.1.1 Détruire la Primitive Sélectionnée

Cette commande permet de détruire la primitive sélectionnée. Un raccourci est d'appuyer sur la touche R13.

6.3.1.2 Détruire Tout sauf les Associées

Cette commande détruit toutes les primitives du dessin sauf les primitives associées.

6.3.1.3 Détruire Tout sauf la Sélectionnée

Cette commande permet de détruire toutes les primitives du dessin sauf la primitive sélectionnée.

6.3.2 Eliminer les Primitives cachées

Cette commande permet de placer dans le presse-papier les primitives recouvertes par d'autres primitives. `xjpdraw` utilise une algorithme très grossier et en conséquence cette commande est très lente.

6.3.3 Copier

6.3.3.1 Copier la primitive sélectionnée

Cette commande permet de copier la primitive sélectionnée. Un raccourci est d'appuyer sur R6. La primitive copiée est strictement identique à la primitive sélectionnée, elle la recouvre donc.

6.3.3.2 Copier avec Homothetie

Cette commande permet de recopier la primitive sélectionnée avec un facteur d'homothétie. Choisir le facteur d'homothétie dans la fenêtre qui apparaît, en agissant sur le curseur et valider par `Fin`.

6.3.4 Le presse-papier

Toutes les primitives détruites au cours d'une session sont placées automatiquement dans le presse papier et sont sauvées dans le fichier `.macdraw`. On peut donc les récupérer à tout moment.

6.3.4.1 Coller

Cette commande permet de rajouter au dessin une primitive sélectionnée dans le presse-papier. Il faut donc d'abord afficher le presse papier puis sélectionner la primitive. Son raccourci est R7.

6.3.4.2 Afficher le Presse-Papier

Cette commande permet d'afficher les primitives détruites. Si l'on est en mode `Sélection` on peut réaliser cette opération en tapant un "a" dans le canevas, et `CTRL-y` dans les autres modes.

6.3.4.3 Masquer le Presse-Papier

Cette commande permet de revenir au dessin courant après avoir affiché le presse-papier. Si l'on est en mode `Sélection` on peut réaliser cette opération en tapant un "m" dans le canevas, ou un `CTRL-m` dans les autres modes.

6.3.4.4 Vider le Presse-Papier

Cette commande permet d'éliminer les primitives du presse-papier pour gagner de la place: on diminue ainsi le nombre de primitives présentes dans les fichiers. Cette commande peut avoir un temps d'exécution important si le nombre de primitive est élevé. Un effet de bord est qu'après cette commande la gomme n'agit plus sur les textures faites avec des lignes (mais vous pouvez détruire les lignes de texture une à une). Un raccourci pour vider le presse papier est `CTRL--`. Pour savoir combien il y a de primitives dans le presse-papier utilisez `CTRL-u` qui donnera le nombre dans la fenêtre d'information.

6.3.4.5 Permuter le presse-papier et les primitives visibles

Cette instruction permet de placer dans le presse-papier toutes les primitives visibles et réciproquement de rendre visible toute les primitives du presse-papier. Elle permet par exemple de créer deux dessins à partir d'un dessin initial.

6.3.5 Associer-Dissocier

6.3.5.1 Associer Tout

Cette commande associe toutes les primitives du dessin courant. Si l'on est en mode "Sélection" on peut réaliser cette opération en tapant un "A" dans le canevas. Dans les autres modes le même effet sera obtenu en appuyant sur `CTRL-a`.

6.3.5.2 Associer les primitives du même type

Cette commande permet d'associer ou de dissocier sélectivement des primitives du même type (par exemple tous les textes). Dans la fenêtre qui apparaît vous pouvez choisir le mode association ou dissociation et dans une fenêtre de choix le type de primitive.

6.3.5.3 Dissocier Tout

Cette commande dissocie les primitives associées et redessine le dessin complet. Si l'on est en mode "Sélection" on peut réaliser cette opération en tapant un "D" dans le canevas, dans les autres modes en utilisant `CTRL-d`.

6.3.5.4 Dissocier la Primitive Sélectionnée

Cette commande dissocie la primitive sélectionnée de l'ensemble des primitives associées. Si l'on est en mode "Sélection" on peut réaliser cette opération en tapant un "d" dans le canevas.

6.3.5.5 Associer tout à .. de la primitive

Cette commande permet d'associer toutes les primitives placées d'une certaine manière par rapport à la primitive sélectionnée. Les possibilités sont à droite, à gauche, au dessus, au dessous. Ce choix est indiqué en cliquant dans un des boutons de la fenêtre qui apparaît.

6.3.6 Sélection Sélective

Cette commande permet de sélectionner par la suite uniquement des primitives du même type. Après déclenchement de cette commande apparaît une fenêtre qui permet de choisir dans une fenêtre de choix le type de primitive que l'on va sélectionner. Un message apparaît en permanence dans le coin supérieur gauche du canevas lorsque l'on est dans ce mode, indiquant le type de primitive que l'on peut sélectionner.

Pour sortir de ce mode choisir `sortir de ce mode` dans la fenêtre de choix. Ce mode présente un intérêt si le dessin compte un grand nombre de primitives.

6.3.7 Changer la valeur de Unitlength

En standard le canevas de `xjpdraw` représente une page de 14.5cm par 14.5cm. Cette commande permet de modifier cette valeur et de faire ainsi un dessin sur une page de taille différente. La taille du dessin est simplement multipliée (sauf pour les textes) par la valeur donnée à l'aide d'un curseur. Depuis la version 3.2 la valeur de `unitlength` est sauvée dans le fichier `.macdraw`. On retrouve alors en rechargeant son dessin la taille qu'il avait lorsque on l'a sauvé. Cette valeur apparaît aussi dans le fichier Postscript.

A part cas particulier l'utilisation de cette commande n'est pas recommandée. Nous verrons par la suite un manière plus élégante de procéder (page 50).

6.3.8 Formule

Si l'on utilise cette commande on va voir apparaître une fenêtre où l'on demande une formule, un valeur de départ `X0`, une valeur de fin `X1`, un pas et la position du point de départ. Ce mode permet de dessiner à l'écran une fonction du type y=f(x) tracée de manière discrète sous la forme d'une ligne polygonale à partir du point de départ pour une valeur de x dans l'intervalle `[X0,X1]`, c'est-à-dire pour les valeurs `X0`, `X0+pas` etc... La formule doit se donner sous la forme $y = f(x)$ comme par exemple:

```
y=sin(x)
y=0.3*sin(3*(log(x)+1)-2)
```

Toutes les valeurs sont données en centimètre. Pour utiliser ce mode vous devez:

- donner la forme analytique de la fonction comme par exemple:

  ```
  y=sin(x)
  y=0.3*sin(3*(log(x)+1)-2)
  ```

- donner les valeurs extrêmes de x (`X0, X1`)
- donner le pas
- donner la position à l'écran du point (`X0, f(X0)`)
- appuyer sur le bouton `Fin`

Les opérations reconnues dans ce mode sont $\times, /, +, -, **$ et les fonctions connues sont *exp*, *log*, *log*10, *sin*, *cos*, *tan*, *atan*, *asin*, *acos*, *sinh*, *cosh*, *tanh*, *sqrt*. De plus la constante π est connue sous la forme `pi`.

La figure 6.1 présente le tracé de la fonction:

```
y=1.5*sin(x**2)
```

pour `X0=0`, `X1=5`, `pas=0.01`, le premier point étant placé à (3,10).

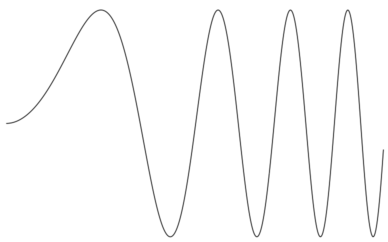


FIG. 6.1 - *Tracé de la fonction* $y = 1.5\sin(x^2)$

Si l'on ne définit que la formule et la valeur de `X0` la valeur de la fonction s'affiche dans la mini-fenêtre d'information: on peut donc se servir de ce mode comme une calculette simple. Si vous n'avez pas défini `X0` on supposera qu'il a la valeur 0. Par contre si vous avez défini une des autres valeurs mais pas toutes vous obtiendrez un message d'erreur et il faudra relancer la commande.

Dans la fenêtre apparaît aussi une fenêtre de texte où l'on peut indiquer un nom de fichier correspondant à un programme (voir le chapitre "Programmation").

6.3.9 Aligner

Pour aligner des primitives par rapport à une primitive fixe il suffit d'associer les primitives voulues en commencant par la primitive fixe, puis d'exécuter la commande `Aligner`. Une fenêtre apparaît et les boutons qui apparaissent indiquent quelle partie de la primitive mobile on va aligner, le nom entre parenthèses décrivant la nature du mouvement que la primitive mobile va subir (horizontal ou vertical). Le menu associé aux boutons permet de stipuler quelle partie de la primitive fixe est utilisée comme référence. Lorsque l'on utilise un ordre où figure la mention de marge il faut savoir que la marge est de 1 mm.

Le bouton `Ajuster la marge` permet d'ajuster la valeur de la marge (en la donnant dans la fenêtre en bas à droite en 1/10 de centimètre). Si on met une valeur nulle le menu des boutons d'alignement ne contiendront plus d'items mentionnant des marges.

6.3.10 Remettre ou enlever Bordure

Ces deux commandes permettent d'enlever ou de mettre une bordure aux primitives associées qui ont été texturées en niveau de gris ou en couleur. Un raccourci pour cette commande est `CTRL-b`. .

6.3.11 Adoucir le polygone

Après sélection d'un polygone cette instruction permet de créer une ligne polygone douce qui passe par les sommets du polygone. Les sommets du polygone sont utilisés comme point de contrôle d'une courbe spline (figure 6.2).

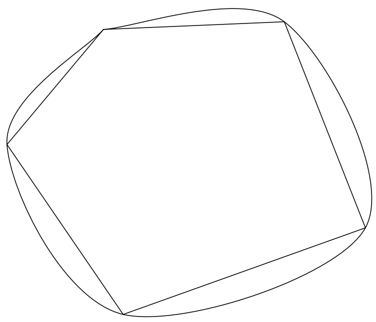


FIG. 6.2 - *Un polygone et son adouci*

6.3.12 Créer un polygone

Cette instruction permet de créer un polygone à partir des droites associées. Le polygone est crée en chaînant les droites dans l'ordre où elles ont été sélectionnées et leur extrémité dans l'ordre fin d'un segment chaîné vers le début du segment suivant. Si l'on choisit `avec ordre de proximité` le chaînage se fait toujours dans l'ordre de l'association mais ce sont les extrémités les plus proches qui sont reliées (figure 6.3).

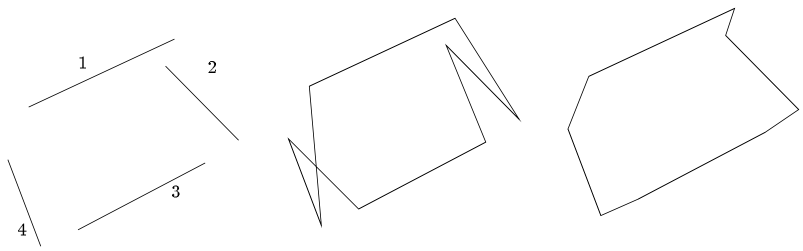


FIG. 6.3 - *Les lignes sont associées dans l'ordre indiqué. A droite les polygones obtenus sans et avec ordre de proximité.*

6.3.13 Dupliquer progressivement

Pour utiliser cette commande vous devez d'abord associer deux primitives de même type. Cette instruction permet de créer n primitives du même type dont la configuration passe progressivement de celle de la première primitive associée à la seconde. Le nombre n est demandé dans la fenêtre de texte. Un exemple de l'utilisation de cette commande est donnée dans la figure 6.4 et à la page 123. Noter qu'il n'est pas possible de dupliquer les polygones. Pour annuler

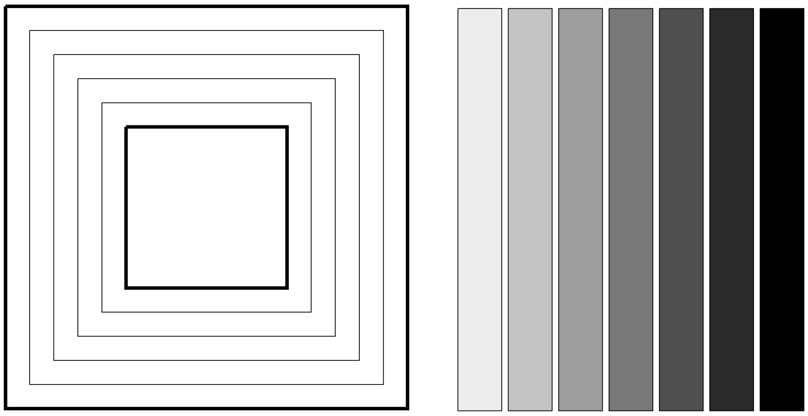


FIG. 6.4 - *Utilisation de l'instruction dupliquer progressivement. Le carré extérieur et le carré le plus à l'intérieur ont été associés et on a demandé de dupliquer 4 fois. A droite on a d'abord crée le rectangle gris clair à gauche puis le rectangle noir à droite.*

cette commande il suffit de donner une chaîne vide dans la fenêtre où l'on demande le nombre de duplication.

6.3.14 Ajouter une flêche au polygone

Cette instruction permet d'ajouter une flêche à une extrémité d'une ligne polygonale. La flêche est ajoutée à l'extrémité la plus proche de l'endroit cliqué au moment de la sélection. Un raccourci pour cette instruction est d'utiliser les touches "f" ou "F" en mode sélection. Cet ordre fonctionne aussi pour les lignes.

6.3.15 Découper les polygones associés selon leur arêtes

Cette instruction permet de découper les polygones associés en segments qui correspondent à leurs arêtes. Les polygones associés passent dans le presse-papier et reste associé. On vérifie préalablement si par cette manipulation on ne crée pas trop de primitives. Si c'est le cas l'opération n'a pas lieu et l'on renvoit un message d'erreur.

6.4 Le Menu `STYLE`

Ce menu permet de sélectionner le style des textes. On a les choix: standard (roman), gras, italique et souligné. . Le style courant est indiqué par un `X`. Vous pouvez aussi indiquer directement le style dans le texte.

6.5 Le Menu `CARACT.`

6.5.1 8pt...24pt

Ces commandes permettent de sélectionner la taille des caractères (8, 10pt par défaut,24,18,14,12 ou 10). La taille courante est indiquée par un `X`.

6.5.2 Texte existant

Avant d'utiliser cette commande vous devez spécifier un style et une taille pour les textes. Cette commande permet de changer la taille et le style des textes déjà entrés de quatre manières:

- `Pour tous`: tous les textes du dessins se dessinent alors dans le style et la taille choisis.
- `Pour les Associées`: seules les primitives associées changent de style et/ou de taille.

– `Pour la Primitive Sélectionnée`: changement de la sélectionnée uniquement

– `Pour la Dernière Primitive Entrée` changement pour la dernière primitive de texte entrée

6.5.3 Recalculer la taille des textes

Cette opération a pour but de calculer la taille exacte de tous les textes visibles en lancant une compilation LaTeX. La boîte englobante des textes indique alors les dimensions. C'est une opération qui peut être relativement longue si le nombre de texte est important (depuis la version 4.2 ce temps a été considérablement réduit car tous les textes sont compilés ensemble avec un tâche de fond). La fenêtre "fast info" indique le nombre restant de primitive à traiter. Cette opération, ainsi que la suivante sont dépendante de votre installation et peuvent ne pas être disponible (voir le chapitre "Installation").

6.5.4 Recalculer la taille des textes associés

Même opération que précédemment mais uniquement sur les textes associés.

6.5.5 Activer la compilation immédiate

Cette commande permet, si vous utilisez le mode compilation immédiate des textes (l'encombrement exact des textes est calculé par compilation des textes en LaTeX), de supprimer les compilations, ce qui permet un léger gain de temps lors de la création des textes (mais l'encombrement des textes indiqué par la boite englobante est alors incorrect). Pour calculer l'encombrement exact il faudra alors utiliser la commande `Recalculer la taille des textes`.

Si vous êtes dans le mode où l'on ne compile pas les textes cette commande permet d'imposer le mode compilation immédiate.

6.6 Le Menu PAGE

Les commandes de ce menu concerne la présentation générale du dessin à l'ecran.

6.6.1 Redessiner Tout

Cette commande redessine le dessin complet, son raccourci est F5. En cas de mauvais fonctionnement de `xjpdraw` l'utilisation de cette commande permet souvent de remettre le programme dans un état convenable.

6.6.2 Règles

Cette commande permet d'afficher ou de masquer dans le canevas des règles graduées en centimètre ou en pixel. Dans ce mode un texte dans la mini-fenêtre indique à tout moment les coordonnées de la souris en centimètre ou en pixel ainsi que les dimensions, le style et la couleur de la primitive sélectionnée dans la fenêtre "fast info".

Si l'on est en mode "Sélection" on peut réaliser cette opération en tapant un "r" dans le canevas (cette opération affiche et masque les règles alternativement). Dans les autres modes on peut utiliser `CRL-r`.

6.6.3 Taille

6.6.3.1 Taille du Dessin

Cette commande affiche dans la mini-fenêtre la position du dessin dans la page, les dimensions actuelles en cm et les dimensions en pixel. Un raccourci pour cette commande est `CTRL-o`.

6.6.3.2 Taille Réelle, Ecran

Le canevas complet représente une page de 14.5cmx 14.5cm dans un canevas qui a une taille de 1000x 1000 pixels. Mais en standard on ne voit qu'une partie de cette page. `Taille Réelle` permet d'avoir sur l'écran la page complète, environ à la même taille que sur le papier. `Taille Ecran` permet de revenir au format standard. Attention l'enregistrement du document doit toujours se faire en taille standard.

6.6.4 Quadrillage

Cette commande permet de faire apparaître ou disparaître un quadrillage dans le canevas. Un raccourci est `CTRL-q`.

6.6.5 Modifier les dimensions du quadrillage

Cette commande permet de changer l'espacement entre les lignes du quadrillage. Donner une valeur en cm dans la fenêtre de texte.

6.6.6 Changer le temps entre les sauvegardes

Régulièrement `xjpdraw` crée un fichier de sauvegarde du dessin courant au format `.macdraw` dans un fichier dont le nom commence par #C suivi du nom du fichier courant. Le temps entre deux sauvegardes automatiques peut être modifié à l'aide de la commande `Changer le temps entre sauvegarde`. Ce temps est aussi automatiquement ajusté en fonction du nombre de primitives. Noter aussi qu'en cas de problème il est aussi possible de récupérer le dessin en utilisant un des fichiers produit par le mécanisme d'annulation d'ordre, voir page 112.

6.6.7 Zoom

Cette commande permet de diminuer ou d'agrandir un dessin. Dans la fenêtre qui apparaît un curseur permet de fixer le rapport d'agrandissement (Figure 6.5). On peut faire un zoom sur un seul axe ou seulement sur des primitives associées. Pour cela le menu du bouton sur le côté du curseur donne clairement les directives. Par défaut on zoome sur tout le dessin dans les deux dimensions. Attention le zoom conserve la structure topologique des primitives, donc un zoom en x seulement ne transforme pas un cercle en une ellipse mais modifie seulement le rayon du cercle. On

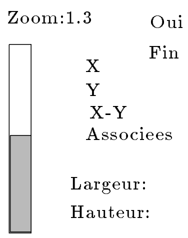


FIG. 6.5 - *la fenêtre de zoom*

valide le zoom par `Oui` et l'on sort par `Fin`.

Le zoom est équivalent à un changement de la taille de la feuille de papier équivalent au canevas. Si l'on zoome tout le dessin `xjpdraw` centrera le dessin dans le canevas après le zoom. Si l'on zoome seulement les associées le centre du rectangle englobant les primitives associées restera invariant. Normalement le facteur de zoom maximum est ajusté pour que le dessin ne puisse pas sortir du canevas. Si toutefois une primitive sort du canevas utiliser la commande `Ramener une primitive hors cadre` du menu `PAGE`. On peut aussi donner dans la fenêtre de zoom les dimensions finales que l'on souhaite pour le dessin. Donner les dimensions voulues (en cm) et tapez retour chariot.

On peut obtenir le même effet de zoom sans changer le fichier source. Par exemple en placant l'instruction suivante devant la commande input de votre dessin:

```
\unitlength 0.5cm
```

on diminuera de 2 la taille du dessin. Pour que cela fonctionne bien il est nécessaire d'avoir enregistré le dessin avec une valeur de unitlength de 1. Une autre possibilité est de forcer la dimension maximale du dessin à être une quantité donnée. Pour cela il suffit d'affecter une valeur négative à `unitlength`, la valeur absolue étant celle affectée à la dimension maximale.

6.6.8 Centrer

6.6.8.1 Centrer tout, les associées

Les commandes `Centrer Tout` et `Centrer les Associées` ont pour effet de placer le dessin si possible au centre du canevas visible sans bouger les glisseurs ou, à défaut, dans le canevas complet. Si cela n'est pas possible (car une primitive sort du cadre) le programme avertit. La solution est alors de diminuer la taille du dessin par un zoom pour rendre ces dimensions compatibles avec la taille du canevas. Ce phénomène peut se produire si un zoom trop fort a été effectué. Si l'on est en mode "Sélection" on peut réaliser un centrage de l'ensemble des primitives en tapant un "c" dans le canevas, ou un `CTRL-c` dans tous les modes.

6.6.8.2 Centrer un texte

On peut aussi centrer ou décentrer un texte dans une primitive. Pour cela il est nécessaire d'associer le texte et la primitive puis d'utiliser la commande `Centrer le texte` (seules les deux primitives doivent être associées). Le centrage peut n'apparaître qu'après réaffichage du dessin. Une primitive de texte centrée à un petit "+" au dessus. On peut centrer un texte dans les rectangles, les cercles, les ovales et les ellipses. On peut aussi bien sûr décentrer un texte. Un raccourci pour cette commande est `CTRl-t`.

6.6.9 Ramener une primitive hors cadre

Cette commande permet de ramener les primitives qui sont hors du canevas dans celui-ci (cela peut se produire lorsque l'on translate des primitives associées).

6.6.10 Compter les textes dans la taille du dessin

Si vous avez des textes assez longs mais qui sont en fait des commandes LaTeX produisant des textes courts et que vous ne compilez pas les textes `xjpdraw` peut être abusé et calculer une mauvaise boîte englobante, ce qui résulte dans un dessin mal centré.

Cette commande permet de ne pas prendre en compte les textes pour le calcul de la boîte englobante fournie à LaTeX pour le calcul de l'encombrement du dessin. Elle n'est donc pas utile si vous compilez les textes.

6.6.11 Mesurer

Cette commande permet de faire apparaître la *boite à mesure* qui va afficher les distances euclidienne, horizontale, verticale et l'angle entre le point courant et le dernier point où l'on a cliqué en mode "Sélection".

6.7 Le Menu DISPOSITION

6.7.1 Plan

Les commandes `Premier Plan`, `Second Plan` permettent de placer des primitives en premier et deuxième plan. Ces commandes n'ont d'effet visible que pour les primitives texturées en niveau de gris ou en couleur. Pour cela il suffit de sélectionner la primitive: elle apparaît alors au premier plan et le restera si l'on sélectionne `Premier plan`. Si l'on sélectionne `Second plan` le dessin complet est redessiné avec la primitive en second plan. Si l'on est en mode "Sélection" on peut réaliser ces opérations en tapant un "p" dans le canevas (premier plan) ou "s" (second plan), `CTRL-s` si l'on n'est pas dans ce mode. Ces commandes sont aussi valables pour les primitives associées.

6.7.2 Pivot

Les commandes `Pivoter a Gauche, a Droite` permettent de faire tourner les figures dans le sens indiqué. Elle correspondent à une rotation de ± 90 degrés autour du centre de la primitive.

6.7.3 Miroir

On dispose de la notion de miroir, c'est à dire que l'on dessine la primitive sélectionnée comme si elle était vue dans un miroir (figure 6.6). On peut placer le miroir au dessus de la figure (miroir horizontal supérieure), en dessous (inférieur), à gauche (miroir vertical gauche) ou à droite (droit). Les primitives associées subissent toutes le miroir.

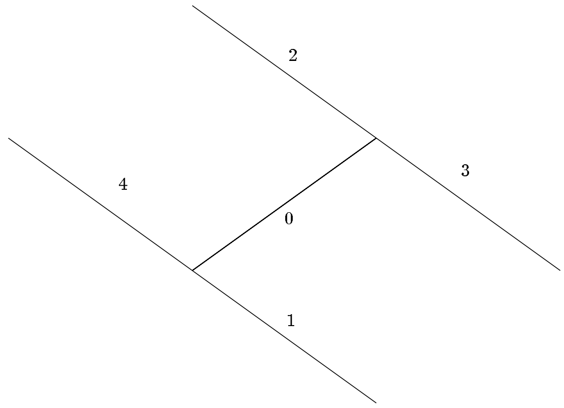


FIG. 6.6 - *Action du miroir: 0=primitive initiale, 1=miroir inférieur, 2=miroir supérieur, 3=miroir droit, 4=miroir gauche*

6.7.4 Réglure

La commande `Réglure` permet de placer un quadrillage invisible dans la fenêtre de dessin. Lorsqu'une primitive est crée, par exemple une ligne, le point de départ ne sera pas celui que vous avez donné mais le nœud du quadrillage le plus proche. Il en sera de même pour l'extrémité de votre segment. Ce mode est évidemment utile pour aligner des primitives dès la création. Par défaut la distance entre deux nœuds du quadrillage est de 0.5cm, mais cette distance peut être ajustée à l'aide d'une commande du menu. Chaque nœuds du quadrillage est indiqué par une petite croix rouge. Un raccourci pour activer/désactiver la réglure est CTRL-i.

6.7.4.1 Aligner la Sélectionnée sur la Réglure

Cette commande permet de redisposer la primitive sélectionnée selon la réglure magnétique:

- pour les lignes: les deux extrémités seront sur un nœud
- pour les rectangles: les quatre coins seront sur un nœud
- pour les polygones: les sommets seront sur des nœuds
- pour les cercles, ellipses: le centre et les sommets de la boîte englobante seront sur des nœuds.
- pour les ovales: la boîte englobante aura ces quatre sommets sur des nœuds.

6.7.4.2 Aligner les Associées sur la Réglure

Cette commande a une action identique à la précédente sur les primitives associées.

6.7.4.3 Aligner Tout sur la Réglure

Cette commande permet de redisposer toutes les primitives déjà crées selon la réglure magnétique.

6.7.4.4 Modifier la réglure

Cette commande permet de modifier les distances entre lignes horizontales et verticales du quadrillage invisible en indiquant à l'aide des curseurs d'une nouvelle fenêtre les nouvelles valeurs, soit en pixel, soit en mm.

6.7.4.5 Supprimer annuler ordre

Il existe dans `xjpdraw` un mécanisme permettant d'annuler jusqu'à la seizième dernière opération effectuée. Ce mécanisme peut sensiblement ralentir la création d'un dessin si le nombre de primitives présentes est élevé puisqu'il repose sur la sauvegarde dans un fichier `.macdraw` du dessin courant après chaque action. Cette commande permet de neutraliser ce mécanisme, qui est d'ailleurs automatiquement supprimé dès que le nombre de primitives du dessin dépasse un seuil (fixé dans le configurateur, voir le chapitre "Installation"). Cette commande permet alors de réactiver le mécanisme d'annulation des derniers ordres, son nom devenant `Activer annuler ordre`.

6.7.4.6 Ordre Inverse

Cette commande permet de dessiner les primitives dans l'ordre inverse de leur apparition à l'écran: les primitives au second plan seront dessinées au premier plan et réciproquement. Seul le tracé est affecté par cette commande, l'ordre initial étant préservé dans le fichier. Pour revenir au dessin initial utiliser la commande `Redessiner tout`.

6.7.4.7 Lignes perpendiculaires, tangentes

Après sélection d'une primitive ces ordres permettent de tracer les deux perpendiculaires ou les deux tangentes à cette primitive au point cliqué lors de la sélection. Les longueurs par défaut des segments crées est de 1.45 centimètre. Un raccourci pour tracer les lignes normales est 'n'.

Notez qu'à l'aide des lignes perpendiculaires il devient relativement facile de tracer une courbe parallèle à une courbe donnée, à une distance fixée. Par exemple pour tracer une courbe parallèle à une courbe spline donnée choisissez d'abord la longueur des lignes perpendiculaires dans le menu `Disposition`. Tracez alors une normale en chaque point de contrôle (pour cela passez en mode "Sélection", cliquez au point de contrôle et utiliser le raccourci 'n'). Passez ensuite en mode "Spline" et activez l'aimant dans le mode "sommet". Cliquez à proximité de la fin de chaque perpendiculaire puis terminer normalement votre nouvelle courbe spline.

6.7.4.8 Longueur des lignes

Cette commande permet d'ajuster la longueur des lignes perpendiculaires et tangentes. Donnez la valeur en cm dans la fenêtre de texte. Cette valeur est fixée par défaut à 1.45 cm.

6.7.4.9 Diviser les lignes associées

Cette ordre permet de découper les lignes associées en tronçons de longueur égale. Le nombre de tronçon est demandé dans la fenêtre de texte.

6.7.4.10 Translation précise

Cette commande permet de translater la primitive sélectionnée (et éventuellement ses associées) d'une quantité fournie au clavier. Donner les valeurs de translations en x et y en cm, séparés par un blanc, dans la fenêtre de texte.

6.7.4.11 Symétrie par rapport à la droite associée

Cette commande nécessite tout d'abord d'associer en premier un segment de droite puis toutes les autres primitives. Ces primitives sont alors transformées dans leur symétrique par rapport à la droite.

6.8 Annuler

Ce bouton permet d'annuler successivement les opérations effectuées dans `xjpdraw` jusqu'a la seizième dernière opération effectuée. Pour cela il suffit de cliquer dans le bouton jusqu'à l'obtention du dessin souhaité. Ce mécanisme peut ralentir `xjpdraw` car il consiste à sauver un nouveau fichier `.macdraw` après chaque opération dans le répertoire précisé par `UNDO_DIR` du configurateur. Il est donc automatiquement supprimé lorsque le nombre de primitive du dessin dépasse un seuil fixé dans le configurateur via la commande `UNDO_NUMBER`. La commande `Supprimer annuler` du menu `DISPOSITION` permet de désactiver ou d'activer ce mécanisme.

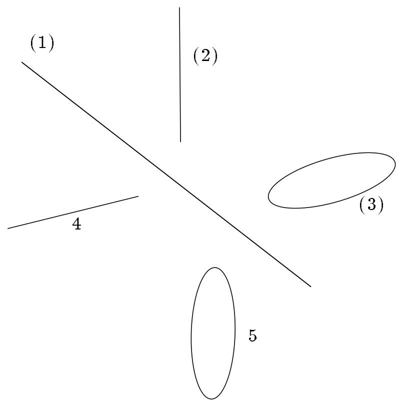


FIG. 6.7 - *On a associé dans l'ordre le segment 1, le segment 2 et l'ellipse 3 (on avait préalablement copié les primitives 2 et 3). On obtient alors les primitives 4 et 5.*

6.9 Le Menu `BIB`

Nous verrons qu'un utilisateur peut *programmer* la création ou la transformation de primitives, à l'aide d'un programme écrit dans un langage dont la syntaxe s'apparente à celle du C. Certains de ces programmes peuvent être d'intérêt général et donc mis à la disposition d'un ensemble d'utilisateurs. Pour ce faire on peut les regrouper dans un répertoire, en leur affectant l'extension `.fi`. Ce bouton avec menu permet d'exécuter un des programmes de ce répertoire (qui doit être installé par la commande `BIB` du configurateur, voir le chapitre "Installation" et le chapitre "Programmation"). La liste des modules disponibles apparaît dans le menu.

En mode "Sélection" il est aussi possible d'exécuter le n^{eme} programme du menu en tapant directement dans le canevas la commande nP (le premier module du menu a le numéro 0).

Il est possible d'avoir des modules communs à un groupe (localisés dans le répertoire indiqué par `BIB` dans votre configurateur) et en plus des modules propres à un utilisateur (à l'aide d'un configurateur local donné par la variable d'environnement `LOCAL_JPDRAW`.

6.10 Le Menu `Externe`

`xjpdraw` a la possibilité de transmettre à un programme des informations sur les primitives existantes et de récupérer graphiquement les résultats de ce programme, qui sera appelé un *module externe*. Le menu attaché à ce bouton donne le nom des modules externes disponibles. Ces modules externes sont définis dans le configurateur sous la forme:

```
# EXTERNE "nom" programme type-primitive type-input type-output
```

`nom` est le nom du module tel qu'il apparaîtra dans le menu. `programme` est le nom du programme qui est cherché tout d'abord dans le répertoire défini dans le configurateur par:

```
# LIB_EXTERNE ...
```

puis dans le répertoire défini dans le configurateur par `LIB`. `type-primitive` est un mot-clé qui permet d'indiquer le type de primitive auquel doit s'appliquer le module externe. Il peut prendre les valeurs:

- `all` : le programme prend en compte l'ensemble des primitives visibles.
- `selected`: seule la primitive sélectionnée est prise en compte
- `associates`: le programme s'applique aux primitives associées.

Pour transmettre les données au module externe `xjpdraw` va les écrire dans un fichier. C'est ce fichier que devra lire le module externe pour traiter les données. `type-input` est le mot-clé qui gère l'écriture des données. Il peut prendre les valeurs:

- `polygon`: les primitives de type rectangle, ovales, cercles, ellipses, arcs de cercles, polygones sont écrites dans un fichier (les textes et les lignes sont ignorés). Chacune des primitives est précédée de la phrase `new polygon` suivi d'une liste de coordonnées (dans le repère page, c'est-à-dire en centimètre). Noter que toutes les primitives transmises sont converties en polygone.

– `typee`: chaque primitive est écrite de manière typée. Les textes sont précédés de `new text` suivi des coordonnées de début du texte et du texte lui même, entre quote. Les lignes sont précédées de `new line` suivie des coordonnées de début et de fin. Les cercles sont précédés de `new circle` suivie des coordonnées du centre et du rayon. Les arcs de cercles sont précédés de `new circle_arc` suivie des coordonnées du centre, du rayon et des angles de début et de fin de l'arc, en degré, orienté dans le sens trigonométrique. Les ellipses sont précédés de `new ellipses` suivie des coordonnées du centre, des longueurs selon les axes x, y et de l'angle de rotation de l'ellipse (en radian).

`type-output` est le mot-clé qui indique sous quel format le module externe sort les résultats. Il peut prendre les mêmes valeurs que `type-input`. Le module externe doit prendre le fichier de données comme argument et sortir les résultats sur la sortie standard (stdout) selon le même format que les données d'entrée. Lorsque un module externe est activé `xjpdraw` écrit le fichier de données, lance l'exécution du module externe avec comme argument le nom du fichier de données et en redirigeant sa sortie sur un fichier temporaire. `xjpdraw` lit alors ce fichier et affiche les résultats.

Par exemple dans le fichier configurateur on peut avoir:

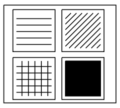
```
# EXTERNE "Intersection des associees" intersection associates polygon polygon
```

dans le but de calculer l'intersection des primitives associées. Le module externe `intersection` est capable de calculer le polygone intersection de plusieurs polygones. On a donc indiqué à `xjpdraw` que les primitives associées devaient être écrite dans le fichier de données sous la forme de polygones et que le résultat est une liste de polygone lue dans le fichier.

Chapitre 7

La texture des primitives

7.1 Texture



Cette icône permet d'activer un mode où l'on va texturer une ou plusieurs primitives existantes (la texture s'applique à l'ensemble des associées). Mentionnons quelques conventions pour la texture:

– une ligne polygonale non fermée est un polygone dont la dernière ligne n'est pas visible à l'écran. Elle est donc texturée comme le polygone fermé

– on ne peut pas texturer les ovales (parce que l'arrondi des ovales n'est, en général, pas connu) mais on peut colorier leur bordure.

– la texture des arcs de cercles est une texture qui s'appuit sur la corde de l'arc.

7.1.1 Comment texturer une primitive

On sélectionne la primitive à texturer puis l'on clique dans l'icône de texture. Une nouvelle fenêtre apparaît: on choisit le type de texture voulu dans cette fenêtre. Cette texture peut être une texture faite de lignes ou un remplissage en gris ou en couleur.
Remarque: Une fois que vous avez sélectionné une primitive puis sa texture vous n'avez plus à repasser par l'icône de sélection pour texturer une autre primitive. Pour sélectionner il suffit de cliquer sur la frontière de la primitive voulue.

7.1.2 Textures faites de lignes

Ces textures sont faites à partir de lignes `xjpdraw` (donc ces textures risquent d'être lourdes en nombre de primitives). Un effet de bord est que la texture n'est pas attachée à la primitive: si vous bougez la primitive la texture ne bougera pas. Pour bouger l'ensemble il faudra associer la primitive et les lignes de texture.

Pour texturer il suffit de cliquer sur le bouton marqué `Oui` à côté de l'icône des textures.

7.1.2.1 Type de texture

Dans `xjpdraw` existent des textures faites de lignes horizontales, verticales, les deux ou bien de lignes inclinées. Ces types de texture apparaissent clairement dans les icônes de la fenêtre. En standard il existe quatre types de texture avec des lignes horizontales, verticales, combinées, où les lignes sont distantes de 0.5cm et enfin une texture avec ligne oblique avec par défaut un angle de 45 degré (figure 7.1). Ces valeurs peuvent être modifiées.

Vous pouvez bien sûr texturer plusieurs fois la même primitive ce qui permet de créer des textures complexes. Il est aussi possible de modifier l'épaisseur des lignes (les lignes de texture sont crées avec le style de ligne courant).

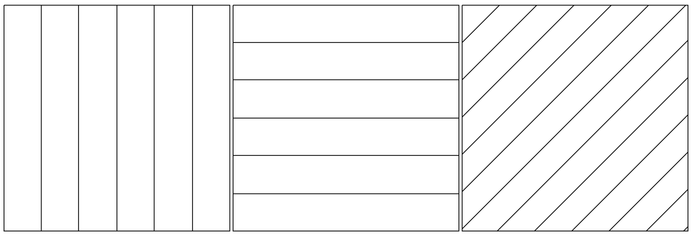


FIG. 7.1 - *Textures standardes de lignes dans un carré*

7.1.2.2 Créer votre texture

Vous pouvez régler la distance entre vos lignes de texture ainsi que leur pente. Pour cela sélectionner l'icône dont l'image contient le message `votre texture` dans la fenêtre associée à l'icône des textures. La nouvelle fenêtre qui apparaît permet de fixer les caractéristiques d'une texture.

Dans cette fenêtre vous avez 3 curseurs: hauteur, largeur, angle. Le premier permet de régler la distance entre des lignes horizontales de texture. Le second la distance entre deux lignes de texture aussi bien pour la texture verticale que pour la texture oblique. Le troisième permet de régler la pente d'une texture, fixée par défaut à 45 degré. Vous pouvez aussi donner le nombre de lignes voulu. Une fois les paramètres de votre texture fixés appuyer sur `Fin`. Pour revenir à la texture standard appuyer sur `Standard`.

7.1.3 Texture à niveau de gris et en couleur

7.1.3.1 Définir une texture

Si on choisit l'icône représentant un carré noirci dans la fenêtre de texture on va mettre une texture à niveau de gris ou en couleur pour la primitive sélectionnée. Après avoir choisi l'icône une nouvelle fenêtre apparaît avec 4 curseurs et des boutons.

Le bouton `Bordure noire` permet de supprimer, conserver ou colorier la bordure de la primitive. Cette fonctionnalité doit être activée avant le coloriage. L'indication `Bordure noire` indique la présence d'une bordure noire (l'intérieure de la primitive sera colorée), `Bordure seule` indique que la texture s'applique à la bordure seule (l'intérieure reste sans texture) et `Pas de bordure` indique que la bordure ne sera pas tracée (l'intérieure sera colorée). Ces différents modes s'obtiennent en cliquant successivement dans le bouton. Après avoir choisi le mode on peut utiliser les curseurs pour définir la texture.

Dans le premier curseur on peut fixer le niveau de gris de la texture entre 1 (noire) et 15 (blanche), voir figure 7.2. La texture blanche apparaît légèrement grisée à l'écran, ceci pour permettre de reconnaître par la suite que la primitive est texturée. A chaque mouvement du curseur la texture apparaît sur la primitive. Le bouton `Fin` permet de valider définitivement la texture.

Les 3 autres curseurs qui apparaissent servent à fixer le niveau de rouge, vert, bleu pour la couleur. Après l'avoir fait on valide la couleur avec le bouton `Fin`.

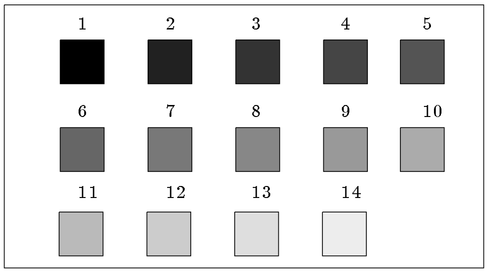


FIG. 7.2 - *Les niveaux de gris*

A noter que vous pouvez directement obtenir la fenêtre des niveaux de gris en utilisant le raccourci `CTRL-g`, la texture s'appliquant sur la dernière primitive crée.

7.1.3.2 Les masques

La texture blanche comporte une propriété intéressante: la partie de toute primitive recouverte par une primitive de texture blanche n'apparaîtra pas sur la feuille de papier. Donc la texture blanche permet de masquer une partie quelconque d'une primitive. On peut donc créer ainsi, par exemple, un arc d'ellipse à partir d'une ellipse complète. C'est ce l'on a fait dans la figure 7.3 à gauche sans supprimer la bordure noire du rectangle qui nous a servi à masquer une partie de l'ellipse et à droite en la supprimant. Noter que l'on peut aussi créer un arc d'ellipse en coupant une

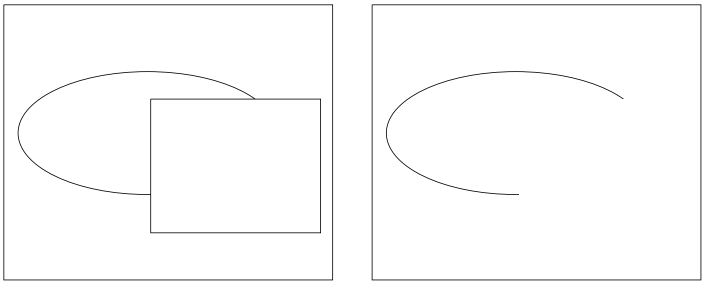


FIG. 7.3 - *L'utilisation d'un masque pour créer un arc d'ellipse*

ellipse avec le ciseau, voir page 33.

7.1.4 La gomme

L'icône dont l'image indique le message `Gomme` permet de faire disparaître une texture dans une primitive. Il suffit de sélectionner cette icône puis de cliquer sur le bouton `Oui` à côté de l'icône des textures. A noter que les textures faite de lignes ne seront effacées que si vous n'avez pas vidé le presse-papier (sinon il faudra détruire les segments un à un).

La texture courante est définie dans l'icône. Si vous n'avez pas sélectionné une primitive la sonnerie retentit.

7.1.5 Les trous dans une texture

Ce qui suit ne concerne que les textures à base de ligne

Si vous texturez une primitive qui englobe d'autres primitives (un polygone qui contient un rectangle par exemple) la texture va remplir complètement le polygone. Il est parfois souhaitable de ne pas texturer les primitives à l'intérieur. Pour cela il suffit d'associer les primitives que l'on ne veut pas voir texturées puis de texturer (Figure 7.4). Cela ne marche toutefois que pour les rectangles, les ellipses, les cercles et les polygones.

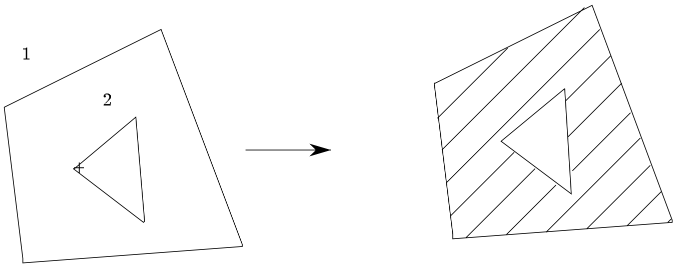


FIG. 7.4 - *Texture avec des trous: 2 est associée (notez le '+' sur sa frontière) et 1 est texturée.*

7.1.6 Rendre une texture permanente

Il est parfois intéressant de définir une texture qui sera utilisée systématiquement pour toutes les primitives qui vont être crées par la suite. Pour cela il suffit de définir la texture voulue pour une primitive puis de la définir comme `Permanente` à l'aide du bouton au-dessus de l'icône des textures. C'est un bouton de choix qui permet de passer soit à `Temporaire` (la texture ne s'applique qu'à la primitive sélectionnée, c'est le défaut), `Permanente` (les primitives crées par la suite auront la texture définie) ou `Definir` qui permet de définir une nouvelle texture que l'on veut rendre permanente. Pour rendre une texture permanente on appuie sur le bouton `Définir`, on texture la primitive, puis on appuie sur le bouton `Permanente`.

Chapitre 8

Le mode transparent

Dans ce mode il est possible de créer des documents de plusieurs pages, ce qui permet entre autre de créer facilement des transparents.

8.1 Première page

Pour initialiser ce mode il suffit de lancer `xjpdraw` de la manière suivante :

```
xjpdraw -slide
```

Dans ce cas la page que présente `xjpdraw` est assez différente de la page classique. Tout d'abord la page représente maintenant une feuille de papier qui fait par défaut 18cm par 23 cm, ces valeurs étant modifiables par le configurateur. Si vous voulez modifier ces valeurs mettez dans le configurateur les ordres du type:

```
# SLIDE_WIDTH 15
# SLIDE_HEIGHT 22
```

qui indique que la largeur d'un picture est de 15cm et sa hauteur 22 cm. Pour faciliter le positionnement des textes le passage en mode transparent fait qu'un grillage est tracé à l'écran et que la réglure magnétique est active (on peut la déactiver/activer par le raccourci CTRL-i, chaque nœud de la réglure est indiqué par une petite croix rouge).

A partir de là vous pouvez créer votre document de manière classique. Il est évidemment possible d'incorporer des fichiers `xjpdraw` classiques avec l'album.

Lorsque l'on est dans le mode transparent une petite fenêtre en bas à droite indique le numéro de page courant ainsi que le nombre total de pages du document. A noter que pour éviter des erreurs de manipulation il n'est pas possible de sortir de ce mode en restant dans `xjpdraw`. De plus recharger un fichier crée en mode transparent fait passer automatiquement `xjpdraw` en mode transparent.

8.2 Les pages suivantes

Une fois la première page crée on peut ajouter de nouvelles pages en utilisant la commande `Nouvelle page` du menu `Disposition`. Les primitives associées dans la page précédente sont automatiquement dissociées.

8.3 Déplacements dans les pages

Pour se déplacer dans le document vous pouvez utiliser la commande `Aller à la page..` de menu `Disposition` ou, en mode Sélection, utiliser les touches `Return` pour avancer à la page suivante ou `backspace` pour reculer d'une page.

8.4 Détruire une page

Cette commande du menu `Fichier` permet de détruire la page courante. Les pages suivantes sont automatiquement renumérotées.

8.5 Ajouter une page

Il est possible d'ajouter une page après la page courante en utilisant cette commande du menu `Fichier`.

8.6 L'album

Lorsque l'on utilise l'album dans le mode transparent il y a deux cas de figure :

– le fichier de l'album est un fichier de transparent: Les pages sont alors ajoutées au document, la première étant incluse dans la page courante. Ainsi pour ajouter les pages de l'album à un document, il faut créer une nouvelle et inclure l'album dans cette page.

– le fichier album est un dessin du mode normal: il est alors simplement ajouté à la page courante.

8.7 Les textes

Dans ce mode les textes subissent un traitement particulier pour rendre ce qui est à l'écran le plus proche possible de ce qui apparaîtra sur la page. L'encombrement du texte et l'espace entre les caractères sont ajustés pour que la largeur du texte à l'écran corresponde à la largeur sur le papier. Il en sera de même lorsque l'on demandera de montrer la page en taille réelle: `xjpdraw` tentera de compresser les textes de façon à rendre leur encombrement le plus proche possible de leur encombrement réel.

8.8 Création d'un document complet

Lorsque l'on sauve un fichier dans ce mode le fichier LaTeX crée contient un environnement picture pour chaque page du document. Mais pour la création de transparent il peut être en plus intéressant d'encapsuler ces environnements dans un cadre LaTeX qui permet de créer une présentation standardisée. Ceci est possible en utilisant le configurateur. Supposons qu'une présentation standarde d'un transparent consiste à avoir un en-tête de document particulière, puis pour chaque page d'encapsuler le texte de la page entre deux instructions qui géreront la présentation générale. Vous allez alors placer dans le configurateur après l'ordre `EN_TETE_SLIDE` la syntaxe de votre préambule, comme par exemple.

```
# EN_TETE_SLIDE  \documentstyle[trans]{article}\n \begin{document}
```

Noter la présence du caractère spécial de retour à la ligne `\n`, qui doit être obligatoirement suivi d'un espace. Vous allez ensuite indiquer avec les ordres `START_SLIDE` et `END_SLIDE` les instructions qui vont encapsuler chaque page, comme par exemple :

```
# START_SLIDE \begin{ltransp}
# END_SLIDE \end{ltransp}
```

Pour créer un document complet il vous suffit d'exécuter alors la commande `Vérification` du menu `FICHIER` pour avoir dans le fichier `bidon.ltex` un document complet, compilable par LaTeX, qui sera du type suivant pour notre exemple :

```
\documentstyle[trans]{article}
 \begin{document}
%quelques ordre speciaux de xjpdraw

%Start of page 1
\begin{ltransp}
\begin{picture}
votre 1ere page
\end{picture}
\end{ltransp}
%End of page 1
....
\end{document}
```

Il est aussi nécessaire d'ajuster la valeur des commandes `SLIDE_WIDTH` et `SLIDE_HEIGHT` pour que les environnements picture soient compatibles avec le style qui est défini dans le préambule.

8.9 Les différences avec le fonctionnement normal

La principale différence avec le fonctionnement normal réside dans les opérations qui agissent sur l'ensemble des primitives comme par exemple `Associer tout`. Dans le mode transparent ces transformations ne vont agir que sur les primitives de la page courante. Il y a cependant une exception, le presse papier, dans lequel figure toutes les primitives détruites de toutes les pages. Cela permet de copier des primitives d'une page pour les reporter dans une autre page: pour cela il suffit d'associer les primitives à copier, de les dupliquer puis de détruire les associées. On se rend ensuite dans la page où l'on désire les mettre, on affiche le presse-papier pour insérer les primitives voulues.

Chapitre 9

Langage: les fonctions utilisateur

Un des défauts des logiciels de dessins existant est l'impossibilité d'interagir autrement qu'avec la souris. Une des motivations de la création de `xjpdraw` est justement de proposer des solutions à ce problème. Nous avons déjà vu une solution: l'usage des modules externes permet de faire traiter les données existantes dans `xjpdraw` par un programme externe. Dans les deux chapitres suivant nous allons introduire deux autres méthodes permettant d'interagir avec `xjpdraw`. Dans le chapitre suivant nous allons décrire un langage de programmation graphique similaire au langage C et dans ce chapitre nous décrivons un langage reposant essentiellement sur des données numériques, celui des *fonctions utilisateur*.

Dans ce langage un dessin est décrit dans un fichier sous une forme purement numérique avec une syntaxe simple qui permet de l'interfacer très facilement avec n'importe quel programme externe. Par exemple ce langage permet très facilement de tracer des courbes planes calculées par un programme externe. Pour cela il suffit de créer un fichier commencant par:

```
mode polygone
-1 10
```

suivi des coordonnées x, y de votre courbe, séparées par un espace, un couple par ligne. Il est alors possible d'indiquer à `xjpdraw` de charger ce dessin et éventuellement de tracer des axes, un grillage etc..

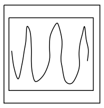
9.1 Fonction utilisateur

9.1.1 Introduction

Traiter des données purement numériques (par exemple des données représentant une courbe plane) est difficile avec les logiciels de dessins existant. Ce problème peut être résolu à l'aide de logiciels spécialisés comme `gnuplot` qui permet de sortir le résultat sous la forme d'un fichier Postscript. Mais ces outils ont des inconvénients:

- ils sont orientés vers l'analyse numérique et manque donc de généralité.
- il est impossible d'éditer le résultat
- les différences entre les fontes utilisées dans le dessin et celle du corps du texte nuisent à la qualité typographique du document

9.1.2 Usage des fonctions utilisateurs



Le mode associé à cette icône est de permettre de tracer des fonctions ou toutes autres figures définies par un utilisateur dans un fichier en utilisant un langage spécifique. C'est un interface puissant et simple entre un programme utilisateur et `xjpdraw`.

Pour les fonctions utilisateurs les axes sont définis comme étant en x croissant selon la largeur de la page et en y croissant du bas de la page vers le haut.

9.1.3 Comment charger un fichier utilisateur

Pour charger une fonction utilisateur il faut d'abord définir et sélectionner un rectangle qui va contenir le dessin. Ce rectangle sera utilisé pour mettre à l'échelle le dessin. Vous n'avez donc pas à vous préoccuper des unités utilisées: vous travaillez dans vos unités et le système s'arrange, c'est-à-dire qu'il gére les changements d'échelle, de façon à ce que le dessin occupe l'intégralité du rectangle. Après sélection du rectangle cliquer dans l'icône fonction pour faire apparaître une nouvelle fenêtre contenant trois icônes. La première sert à entrer la fonction, la deuxième à faire tracer des quadrillages, la troisième à ajuster les paramètres du quadrillage.

Cliquer dans la première icône: une nouvelle fenêtre apparaît. Donner le nom de votre fichier fonction (terminé par un `Return`) dans la première fenêtre de texte.

Si vous n'avez pas commis d'erreur de manipulation vous devriez voir alors apparaître votre dessin.

Deux autres manières de procéder sont possibles. La première consiste à indiquer la fonction utilisateur au lancement du programme. La syntaxe est alors:

```
xjpdraw -X x1 -Y y1 -F nom [-E xmin xmax ymin ymax][-O x2 y2]
```

où `x1, y1` représente la largeur, hauteur en cm de la boite où l'on va tracer la fonction. `nom` est le nom du fichier fonction (une variante est de mettre `-Fp` à la place de `-F`, dans ce cas on tracera pas le cadre englobant mais il sera toutefois présent dans le presse-papier). Les arguments optionnels sont:

1. `-E xmin xmax ymin ymax` qui permet de tracer le dessin avec une échelle fixée. Les arguments représentent les coordonnées minimum et maximum de la fonction dans la boite. Les arguments de `-E` sont en unité de la fonction.

2. `-O x2 y2` qui permet de fixer la position du coin supérieur gauche de la boite (par défaut il est placée à 2cm en x et y du coin supérieur gauche du canevas). Les arguments de `-O` sont en cm.

La troisième manière de charger une fonction utilisateur est de le faire à partir d'un fichier de programmation, se référer au chapitre "Programmation".

9.1.4 Les fichiers utilisateurs: le langage

Les primitives dessinées à l'aide du mode fonction sont décrites dans un fichier utilisateur en utilisant la notion de `mode`. Un mode décrit une primitive ou une manière de relier les points contenus dans le fichier utilisateur.

Un fichier utilisateur peut commencer par le nombre de lignes du fichier (ceci n'est nécessaire que si le fichier fonction ne commence pas par le mot `mode`) puis par la description des primitives, c'est-à-dire la donnée de modes suivie de données numériques. Nous allons maintenant décrire les différents modes.

9.1.4.1 Le mode comment

Ce mode permet de placer des commentaires dans la fonction utilisateur. Le commentaire se termine par un #.

Exemple

```
mode comment
Voila un commentaire utile
#
```

9.1.4.2 Le mode connecte

Ce mode permet de tracer une ligne polygonale dont les coordonnées des sommets sont les données numériques. Le style de la ligne polygonale est défini par le mot-clé qui suit le mot `mode` selon le tableau suivant:

standard	vecteur	pointillé
connecte	connecte_vecteur	connecte_pointille
connecte_epais	connecte_vecteur_epais	connecte_pointille_epais

Les segments seront épais en mode `connecte_epais` et des vecteurs en mode `connecte_vecteur`. Le mode `connecte` est le mode par défaut.

Exemple: Dans cet exemple on trace une rampe commencant en (0,0), passant par (1,1) et horizontal à partir de ce point jusqu'a (1,2).

```
mode connecte
0 0
1 1
1 2
```

Si un grand nombre de points sont à tracer il est préférable d'utiliser le `mode polygone`.

9.1.4.3 Le mode deconnecte

Ce mode permet de tracer un ensemble de segments. Les données numériques qui suivent ce mode définissent les coordonnées des extrémités des segments, sous la forme:

```
mode deconnecte
x1 y1
x2 y2
.. ..
```

Comme pour le mode connecte le style des segments dépend du mot-clé suivant le mot `mode`:

standard	vecteur	pointillé
deconnecte	deconnecte_vecteur	deconnecte_pointille
deconnecte_epais	deconnecte_vecteur_epais	deconnecte_pointille_epais

Le cas précédent peut ainsi s'ecrire:

```
mode deconnecte
0 0
1 1
1 1
2 1
```

9.1.4.4 Le mode cercle

Ce mode permet de rajouter un cercle ou une ellipse dans le dessin. La syntaxe est:

```
mode cercle
x,y
rx ry
```

où `x, y` sont les coordonnées du centre du cercle ou de l'ellipse et `rx ry` les dimensions selon les axes x, y (donc égales si c'est un cercle). Les syntaxes possibles pour le style sont:

standard	pointillé
cercle	cercle_pointille
cercle_epais	cercle_pointille_epais
cercle_noir	

Le cercle sera noirci en mode `cercle_noir`.

Pour les dimensions des cercles il est possible de les donner en cm ou mm à la place de les exprimer en unité utilisateur. Dans ce cas le cercle n'est pas utilisé pour calculer l'encombrement du dessin. Il n'est donc pas possible de créer un fichier ne contenant que ce type de données (pour résoudre ce problème il suffit d'englober le dessin dans un rectangle).

Par exemple pour tracer un cercle de rayon 1 unité centré en 0,0 le fichier contiendra:

```
mode cercle
0 0
1 1
```

On peut aussi indiquer un angle de rotation pour les ellipses. Pour cela il suffit de mettre comme dimension de l'ellipse un nombre négatif et de faire suivre les dimensions de l'angle de rotation en degré. Ainsi pour tracer une ellipse centrée en 0,0 , de grand axe 100, petit axe 50 inclinée à 45.5 degré:

```
mode cercle
0 0
-100 50 45.5
```

9.1.4.5 Le mode arc

Ce mode permet la définition d'arc de cercle. La définition est:

```
mode arc
x,y
rx ry
angle1 angle2
```

où `angle1 angle2` sont les angles en degré compris entre 0 et 360 degrés qui définissent l'arc (en tournant toujours dans le sens trigonométrique). L'arc est défini comme partant du point défini par `angle1` et allant vers le point défini par `angle2`.

Les syntaxes de ce mode sont:

standard	pointillé
arc	arc_pointille
arc_epais	arc_pointille_epais
arc_noir	

Exemple

```
mode arc
1 7
1 1
10 200
```

définit un arc sur un cercle centré en 1,7 de rayon 1 commencant à 10 degrés et terminant à 200. On obtient alors l'arc de la figure 9.1. Si on avait inversé les valeurs 10 200 on aurait obtenu l'arc complémentaire.

9.1.4.6 Le mode croix

Ce mode permet de tracer à l'écran des croix de dimensions indépendantes des unités utilisateur. Sa syntaxe est:

```
mode croix
x y
largeur hauteur
```

où `x, y` représente le centre de la croix, les dimensions étant données soit en mm ou en cm, comme, par exemple `2mm 0.5cm`.

9.1.4.7 Le mode texte

Le mode `texte` permet de placer une chaîne dans le dessin. Pour chaque texte on met dans le fichier fonction:

```
mode texte
x y
type chaine
```

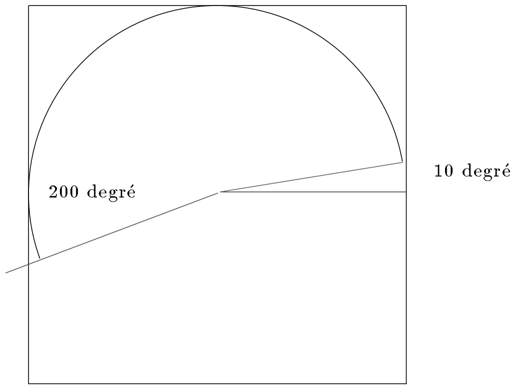


FIG. 9.1 - *Un arc de cercle*

`x y` définit la position du coin inférieur gauche du texte, `chaine` est le texte à écrire. `type` indique le style du texte. C'est un entier dont le chiffre des unités indique la taille des caractères (0=10pt, 1=12pt, 2=14pt, 3=16pt, 4=18pt, 5=24pt) et le chiffre des dizaines le style (0=standard, 1 =gras, 2=italique). Si ce nombre est négatif le texte sera de plus souligné. La chaîne ne doit pas compter plus de 256 caractères et ne doit pas contenir de newline. Dans le type on peut indiquer aussi un angle de rotation du texte (en degré). La convention reste identique pour le style et la taille du texte. Le chiffre des milles et des centaines permet d'indiquer un angle de rotation en degré. Si de plus un chiffre est ajouté pour les 10 000 alors l'angle de rotation est négatif.
Exemples:

11 : gras, 12 pt -12: gras, 14pt, souligné
4511 : gras, 12pt tourné à 45 degrés
14511: gras, 12pt tourné à -45 degrés

```
mode texte
0 5
11 here we get a bold, 12 pt string
```

9.1.4.8 Le mode rectangle

Le mode `rectangle` permet de tracer un rectangle dans la figure. La syntaxe est:

```
mode rectangle
x y
largeur hauteur
```

où `x y` sont les coordonnées de l'origine du coin inférieur gauche du rectangle. **Exemple**:

```
mode rectangle
0 0
1 2
```

Le mot suivant `mode` permet de modifier le style des lignes utilisées pour tracer le rectangle:

standard	pointillé
rectangle	rectangle_pointille
rectangle_epais	
rectangle_noir	

Si les dimensions sont données avec des unités comme le mm ou le cm alors la position donnée est le centre du rectangle. Un exemple est:

```
mode rectangle
0 0
0.3mm 0.3mm
```

9.1.4.9 Le mode polygone

Le mode polygone permet de produire des polygones ou des lignes polygonales. La syntaxe est:

```
mode polygone
nb type
x1 y1
x2 y2
```

`nb` est le nombre de sommet du polygone: si ce nombre est inconnu mettez -1 pour `nb`. `type` est un entier décrivant le style et le type du polygone selon le codage suivant:

type	0	1	2
	polygone fin	polygone épais	polygone pointillé
type	10	11	12
	ligne polygonale fin	ligne polygonale épaisse	ligne polygonale pointillée
type	3	13	14..19
	polygone pointillé épais	ligne polygonale pointillée épaisse	ligne polygonale pointillée

Les types 12, 14 à 19 diffèrent simplement par le type de pointillé. Rappelez vous qu'il ne peut pas y avoir plus de 30 points dans un polygone.
Exemple: Dans cet exemple on trace 2 polygones, le premier comportant 5 points, reliés par des lignes fines et le second 4 points reliés par des lignes épaisses.

```
mode polygone
5 0
3 4
5 6
3 2
1 5
2 3
mode polygone
4 1
2 1
2 3
3 2
1 1
```

Utiliser les polygones pour tracer une fonction est une astuce facile pour générer des fichiers importants. En effet le mode connecte ne permet de tracer par défaut que 1500 segments. Par contre un polygone pouvant contenir 30 points, mais ne comptant que pour une primitive, utiliser des polygones permet de tracer des fonctions d'environ 1500x 30points. Si vous avez mis -1 comme nombre de points du polygone `xjpdraw` génèrera autant de lignes polygonales que nécessaire. Par exemple si vous utilisez le fichier suivants:

```
mode polygone
-1   10
x1   y1
.. une liste de 45 points
```

`xjpdraw` va générer alors autant de polygones de 30 points que nécessaire, puis un dernier contenant le reste des points de la liste.

9.1.4.10 Mode bspline

Ce mode permet de créer des courbes de type b-spline en indiquant les coordonnées des points de contrôle. La syntaxe est:

```
mode bspline
n1 n2
x1 y1
x2 y2
.....
```

où `n1` est le nombre de points de contrôle, `n2` le style des traits selon le code du mode polygone et `xi,yi` les coordonnées des points de contrôle.

9.1.4.11 Mode spline

Ce mode permet de tracer des courbes splines. La syntaxe est identique à celle du mode bspline. Mais la courbe passe par les points de contrôle.

9.1.4.12 Le mode epaisseur_trait

Ce mode permet de définir l'épaisseur des traits pour les primitives qui le suive. On indique l'épaisseur en unité métrique de la manière suivante:

```
mode epaisseur_trait 0.1cm
```

où cm peut être remplacé par mm, ou bien on peut les indiquer en unité utilisateur simplement par:

```
mode epaisseur_trait 0.1
```

9.1.4.13 Exemple de fonction utilisateur

Dans cet exemple on trace un tronc de cône vue de dessus.

```
mode polygone
-1 10
-1.000000 0.000000
-0.750000 -0.935414
-0.500000 -1.224745
-0.250000 -1.369306
0.000000 -1.414214
0.250000 -1.369306
0.500000 -1.224745
0.750000 -0.935414
1.000000 0.000000
mode deconnecte
-1.000000 0.000000
-0.500000 4.000000
1.000000 0.000000
0.500000 4.000000
mode cercle
0 4.000000
0.500000 0.707107
mode texte
0 0
0 voici un tronc de cone
```

Dans cet exemple le `mode polygone` est utilisé pour tracer un arc d'ellipse, représentant la base du cône, le `mode deconnecte` pour tracer les côtés du cône et le mode cercle pour tracer l'ellipse représentant le haut du tronc de cône.

9.1.5 Erreur dans les fonctions utilisateurs

Une erreur possible est de tracer un dessin non inscriptible dans un rectangle (par exemple une fonction y=cte). Dans ce cas il y a une double sonnerie et pas de tracé. Une bonne habitude de programmation est de toujours inclure dans un fichier fonction un rectangle englobant, qui permet d'éviter ce genre de problème.

Si `xjpdraw` détecte des anormalités dans le fichier fonction il va émettre des messages à la console. Ces messages sont de deux types:

- des messages d'erreur non fatale: un mode a été trouvé sans que l'on puisse créer la primitive correspondante (pas de point dans un polygone par exemple). Cela n'arrête pas le processus.
- des messages d'erreur fatale: un mode a été incorrectement défini. Le dessin ne sera pas crée.

Dans les deux cas on imprime à la console le numéro approximatif de la ligne incriminée.

9.1.6 Tracé d'abscisses et d'ordonnées

Pour tout dessin crée par une fonction utilisateur il est possible de faire apparaître automatiquement un quadrillage du rectangle avec l'impression des coordonnées des lignes du quadrillage.

Juste après avoir introduit votre fonction cliquer dans la deuxième icône de la fenêtre fonction pour avoir un quadrillage standard. La distance entre les lignes du quadrillage sera de 2 cm et les cooordonnées auront 2 décimales (voir un exemple à la page 121).

Si le type de quadrillage par défaut ne vous convient pas vous pouvez le modifier à votre convenance en sélectionnant la troisième icône de ce menu, juste après avoir introduit la fonction. Une nouvelle fenêtre apparaît où vous pourrez définir les paramètres de votre quadrillage (figure 9.2).

Dans cette fenêtre vous allez définir:

- la distance entre les lignes du grillage (soit en centimètres, soit en unité utilisateur, soit en donnant le nombre de ligne)
- le format de sortie des abscisses et ordonnées qu'il faut donner sous la forme "n.m" où n et m sont des entiers: par exemple 6.2 indique que vous voulez deux chiffres après la virgule et un total de 6 caractères (`xjpdraw` vous proposera des formats à partir de l'analyse des données).
- éventuellement vous pouvez aussi indiquer le début des premières lignes horizontales et verticales en indiquant leur coordonnées en unité utilisateur dans les fenêtres "1ere ligne" (ces coordonnées ne doivent pas être plus petite que les minimum de la fonction, qui sont indiqué dans la fenêtre d'information).
- vous pouvez placer aussi les échelles soit en x uniquement, soit en y uniquement, soit en y à gauche ou à droite.
- éventuellement des mini traits sur les bords du rectangle, entre les lignes du quadrillage. Les mini-traits sont simplement des petites graduations entre les graduations principales. Les curseurs des mini-traits permettent de régler la distance entre les mini-traits pour les abscisses et les ordonnées. Si la distance est nulle il n'y a pas de mini-trait (c'est le cas par défaut).

Noter qu'en général les valeurs données dans les fenêtres de texte doivent être validées en tapant `Return`. Par défaut le dessin va être quadrillé totalement, pour faire apparaître juste les marques du quadrillage sur le bord du rectangle choisissez `Pas de quadrillage`.

Une fois que vous avez défini les paramètres du quadrillage vous l'appliquez en cliquant dans la deuxième icône de la fenêtre fonction.

Après avoir chargé une fonction il est possible d'en tracer d'autre sans qu'il y ait changement d'échelle. Pour cela cliquer sur le bouton `PAS DE CHANGEMENT D'ECHELLE` lorsque apparaît la fenêtre d'entrée du fichier fonction. Il est possible de ne pas avoir de changement d'échelle juste sur l'axe x ou l'axe y.

9.1.7 Echelle fixe

Plutôt que d'utiliser le mécanisme de mise à l'échelle automatique vous pouvez fixer les coordonnées maximum et minimum des points contenus dans le rectangle de la fonction. Pour cela, après avoir cliqué dans la première icône de la fenêtre fonction et avant d'introduire le nom du fichier fonction donner dans la dernière fenêtre de texte les maxima et minima des coordonnées des points contenus dans le rectangle (par exemple 0 3 1 4 pour indiquer que le côté gauche du rectangle est à $x = 0$, le côté droit à 3, le côté bas à $y = 1$, le côté haut à $y = 4$). .

Une autre manière de procéder est de donner l'échelle au lancement comme indiquée page 66.

9.1.8 La couleur et les niveaux de gris

9.1.8.1 Couleur

La notion de couleur pour les primitives existe dans le langage. Pour remplir une primitive, ou pour assigner une couleur à une primitive qui n'a pas d'intérieur, on utilise:

```
modeRVB-(rouge)-(vert)-(bleu)
```

à la place de `mode`. `rouge, vert, bleu` sont des entiers compris entre 0 et 255 définissant les niveaux de couleur. Pour une primitive avec intérieur la bordure est alors tracée en noire. Pour ne pas avoir de bordure noire il faut utiliser:

```
modeRVB--(rouge)-(vert)-(bleu)
```

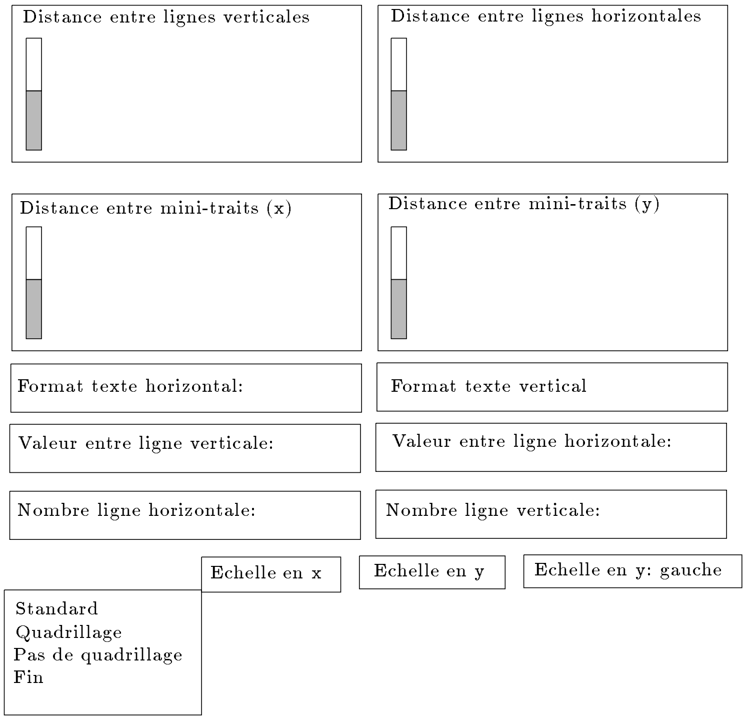


FIG. 9.2 - *La fenêtre permettant d'ajuster un quadrillage*

Pour mettre une couleur uniquement à la bordure utiliser:

`modeRVB+-(rouge)-(vert)-(bleu)`

9.1.8.2 Niveau de gris

Une autre possibilité est la définition de texture en niveau de gris. Pour cela on peut définir le niveau de gris par `modegrisxx` où `xx` est un entier compris entre 1 et 100 qui indique le pourcentage de blanc de la texture (1=noir, 100=blanc). Par exemple `modegris50` donnera une primitive avec une bordure noire et un intérieur en demi-gris.

Si ce nombre est négatif la primitive n'aura pas de bordure. Si le nombre est précédé de `+-` seule la bordure aura le niveau de gris, l'intérieur restant sans couleur.
Exemple:

```
mode connecte
0 0
0 9
9 9
9 0
0 0
modegris90 polygone
-1 0
5 5
3 4
4 2
modegris90 rectangle
0 0
1 2
modegris-95 rectangle
2 2
```

```
1 2
modegris80 cercle
4 4
1 1
modegris-80 cercle
5 5
0.5 0.5
modegris90 cercle
7 7
0.4 0.7
modegris-90 cercle
8 8
0.4 0.7
```

9.2 Résumé des modes

mode	1er argument	2eme argument	arguments suivants
comment	texte..#	-	-
connecte	-	-	x,y
connecte_vecteur	-	-	x,y
connecte_pointille	-	-	x,y
connecte_epais	-	-	x,y
connecte_vecteur_epais	-	-	x,y
connecte_pointille_epais	-	-	x,y
croix	x,y centre	hauteur, largeur (mm ou cm)	
deconnecte	x,y origine	x,y fin	x,y segment
deconnecte_vecteur	x,y origine	x,y fin	x,y segment
deconnecte_pointille	x,y origine	x,y fin	x,y segment
deconnecte_epais	x,y origine	x,y fin	x,y segment
deconnecte_vecteur_epais	x,y origine	x,y fin	x,y segment
deconnecte_pointille_epais	x,y origine	x,y fin	x,y segment
epaisseur_trait	épaisseur[unité]	-	-
cercle	x,y centre	petit,grand axe, angle rotation	-
cercle_pointille	x,y centre	petit,grand axe, angle rotation	-
cercle_epais	x,y centre	petit,grand axe, angle rotation	-
cercle_pointille_epais	x,y centre	petit,grand axe, angle rotation	-
cercle_noir	x,y centre	petit,grand axe, angle rotation	-
arc	x,y centre	petit,grand axe	angle 1, angle 2
arc_pointille	x,y centre	petit,grand axe	angle 1, angle 2
arc_epais	x,y centre	petit,grand axe	angle 1, angle 2
arc_pointille_epais	x,y centre	petit,grand axe	angle 1, angle 2
arc_noir	x,y centre	petit,grand axe	angle 1, angle 2
texte	x,y	style texte	-
rectangle	x,y coin inf. gauche	largeur-hauteur	-
rectangle_pointille	x,y coin inf. gauche	largeur-hauteur	-
rectangle_epais	x,y coin inf. gauche	largeur-hauteur	-
rectangle_noir	x,y coin inf. gauche	largeur-hauteur	-
polygone	nb point(ou -1), type	x,y sommets	-
bspline	nb point contrôle., type	x,y points	-
spline	nb point contrôle., type	x,y points	-
epaisseur_trait	épaisseur des traits	-	-

Chapitre 10

Langage: la programmation

xjpdraw dispose aussi d'un langage de programmation de syntaxe sensiblement identique à celle du langage C avec des variables, des vecteurs, des instructions conditionnelles, des instructions de création de primitives graphiques etc...

C'est ce langage qui est utilisé pour créer des modules assurant des fonctions particulières qui peuvent être ensuite utilisés dans xjpdraw à l'aide du menu BIB. Ainsi si vous avez besoin de tracer une ligne de sinusoïdale il suffit d'écrire un petit programme très simple, de le placer dans le répertoire BIB pour pouvoir l'utiliser continuellement dans xjpdraw. On donne ici l'exemple d'un tel programme permettant de tracer une ligne sinusoïdale entre deux points donnés à la souris:

```
info "Wiggly line,ligne sinusoidale de A a B,En mode selection Droite=abort"
input_mouse 2
%1=distance from %v0 to %v1
%2=atan((%v1[1]-%v0[1])/(%v1[0]-%v0[0]))
%3=0.075 /* amplitude sinusoide */
%4=40 /* periode sinusoide*/
%6=pi/(2*%4)+%1/2 /*u2*/
%5=%6-%1
#0=Formule %3*sin(%4*x), x between %5,%6 , step 0.0075 , start %v0
rotate #0 around %v0 , %2
quit
```

Dans cet exemple on a demandé à xjpdraw d'émettre un message quand le programme est activé (ligne info), d'attendre ensuite que deux points soient entrés à la souris (ligne input_mouse), de calculer la distance entre les deux points, de tracer une sinusoïde horizontale à partir du premier point, de longueur égale à la distance et enfin de tourner la ligne polygone ainsi crée. Ainsi en quelques lignes nous avons pu définir une fonction relativement complexe. Ce langage va maintenant être décrit.

10.1 Introduction

A la différence des fonctions utilisateurs le langage n'utilise pas que des valeurs numériques pour la description des primitives: il repose plutôt sur l'utilisation de variables. L'unité utilisée est le centimètre.

Les programmes écrit dans ce langage sont exécutés en lançant xjpdraw par:

```
xjpdraw -F [nom du fichier programme]
```

On peut aussi lancer le programme en mode invisible, c'est à dire que l'on n'ouvre pas de fenêtre graphique (on peut ainsi lancer xjpdraw d'un terminal quelconque) par:

```
xjpdraw -FI [nom du fichier programme]
```

Si le programme est paramétré (voir plus loin) la syntaxe est:

```
xjpdraw -F [nom du fichier programme][[arguments]]
```

comme par exemple

```
xjpdraw -F essai.f[1,2,3] xjpdraw -F essai.f[tata,toto.ltex]
```

Ces programmes pourront aussi être exécutés en les lancant à partir du menu du bouton `BIB` . Ce sont les programmes que l'on trouve dans le répertoire indiqué dans le configurateur et qui ont l'extension `.fi` . Seuls les programmes non paramétrés sont exécutables.

Enfin on pourra utiliser ces programmes en donnant le nom d'un fichier dans la fenêtre associée à la commande `Formule` du menu `EDITION`.

10.2 Variables et expressions

10.2.1 Expressions élémentaires

Une expression élémentaire est une collection de nombre et d'opérateurs respectant les règles de la programmation en C, comme par exemple:

```
3.14*5.6/(7.8/2)
```

Le nombre π est aussi connu, il s'écrit `pi`.

Les opérateurs élémentaires reconnus sont:

```
- + / * **
```

Certaines fonction mathématiques sont aussi reconnues. Ce sont:

```
exp ceil floor log log10 sin cos tan atan asin acos sinh cosh tanh sqrt rint atan2
```

`rint` retourne la partie entière d'un flottant. `atan2` prend comme argument un vecteur %v et calcule l'arc tangente du rapport %v[1] sur %v[0].

On peut construire des expressions élémentaires par combinaison d'opérateurs, comme par exemple:

```
3.14*sin(pi/3.0)*cosh(3**3)
```

10.2.2 Variables

Dans ce langage une variable commence par un % suivi d'un entier. On peut affecter à une variable la valeur d'une expression élémentaire comme par exemple:

```
%1=3.14*sin(pi/3.0)*cosh(3**3)
```

Vous pouvez aussi utiliser une variable pour construire une expression élémentaire:

```
%2=3.14
%1=%2*sin(pi/3.0)*cosh(3**3)
```

On peut donner un nom à une variable en déclarant préalablement avec la commande `alias_variable` un nom de substitution suivie du numéro de la variable qu'il remplace:

```
alias_variable "toto" 0
```

Par la suite la chaîne {`toto`} pourra être utilisée à la place de la variable `%0` (qui pourra continuer à être utilisée).

10.2.3 Vecteurs

Un vecteur est un tableau de deux flottants. On peut affecter un nom à un vecteur, nom qui commence par `%v` suivi d'un entier. Un vecteur se définit par:

```
vecteur=( [expression elementaire], [expression elementaire])
```

on peut donc écrire:

```
%v1=(3.14*sin(pi/3.0)*cosh(3**3), 3)
%v2=(%10,%11)
```

On peut récupérer les valeurs des éléments d'un vecteur et l'affecter à une variable par:

```
%1=%v1[0]
%2=%v1[1]
```

On peut aussi faire l'inverse:

```
%v1[1]=%2
```

10.3 Instructions conditionnelles

Il est possible de commander l'exécution d'une partie d'un programme seulement si certaines conditions sont satisfaites.

10.3.1 Conditions

Une condition porte sur la valeur d'une ou plusieurs variables et s'écrit selon la syntaxe :

```
condition: ( [variable] [operateur] [variable])
```

où la liste des opérateurs permis est :

- $>$: supérieur
- $>=$: supérieur ou égal
- $<$: inférieur
- $<=$: inférieur ou égale
- $==$: égalité
- $<>$: différent

Un opérateur particulier est la négation:

```
(! [variable])
(! condition)
```

qui renvoit 1 si la variable ou la condition vaut 0 et 0 dans les autres cas.

10.3.2 Combinaison de conditions

Plusieurs conditions peuvent être combinées pour construire une nouvelle condition:

```
condition: condition [operateur] condition
```

où la liste des opérateurs permis est :

- `&&` : et logique
- `||` : ou logique

10.3.3 If

Une instruction conditionnelle peut être écrite avec l'opérateur `if`. La syntaxe de l'instruction `if` est:

```
if condition bloc
     instruction a executer
end_bloc
```

Voici un exemple:

```
if ((%1 <0) && (%2>=0)bloc
if (%0>10) bloc
%1=%1+10
end_bloc
end_bloc
```

où nous avons deux instructions conditionnelles imbriquées.

10.4 Boucle

Les boucles sont utilisées pour exécuter un même bloc de code tant qu'une condition n'est pas satisfaite.

10.4.1 For

La commande `for` peut être utilisée pour définir une boucle. La syntaxe de l'instruction `for` est:

```
for [variable] from [variable] ; condition ; step [variable] bloc
instruction a executer
end_bloc
```

Voici un exemple:

```
for %2 from %10 ; (%2 <= %11) ; step %20 bloc
for {za} from {za1}; ({za} < {za2}) ; step 0.1 bloc
%1=%1+10+pi
end_bloc
end_bloc
```

Dans cet exemple la première boucle est exécutée tant que la variable %2 n'est pas plus grande que la variable %11, la variable %20 étant ajoutée à %2 lorsque le programme atteint la fin de la boucle.

10.4.2 While

Un autre type de commande pour les boucles est l'instruction `while`. dont la syntaxe est:

```
while condition bloc
        instruction a executer
end_bloc
```

Voici un exemple:

```
%0=0
while (%0 <5) bloc
        print(%0,"\n")
        %0=%0+1
end_bloc
```

qui va imprimer la liste des cinq premiers entiers.

10.5 Instruction de branchement

On peut interrompre l'exécution séquentiel d'un programme en indiquant un branchement à une partie de programme. Pour cela il faut placer une étiquette au début de cette partie de programme. Une étiquette est une suite de lettre suivie d'un ":". L'instruction de branchement `goto` est suivie du nom de l'étiquette entre ". Voici un exemple:

```
deb:

while (%0 <5) bloc
        print(%0,"\n")
        %0=%0+1
end_bloc

if (%1==1)bloc
        %1=%1+1
        %0=0
        goto "deb"
end_bloc
```

10.6 Chaîne de caractères

Une chaîne de caractères est incluse entre ". Elle peut contenir les caractères spéciaux. `\n \r \t \b \\`

On peut aussi créer une chaîne de caractères variable en utilisant l'opérateur de concaténation ".":

```
%0=10
"toto".%0
```

et la chaîne vaut `toto10`. Si la variable n'a pas une valeur entière c'est l'entier le plus proche qui sera ajouté à la fin de la chaîne.

10.7 Impression

Imprimer des valeurs et des chaînes sur le terminal de sortie peut être fait avec l'instruction `print` selon une des syntaxes suivantes:

```
print([chaine de caractere])
print([chaine de caractere],[chaine de caractere])
print([chaine de caractere],[variable],[chaine de caractere])
print([chaine de caractere],[vecteur],[chaine de caractere])
print([numero primitive])
```

Noter que certaines séquences de caractères ont un sens spécial:

- `\n` : newline
- `\r` : line feed
- `\t` : tab
- `\b` : backspace

Exemple:

```
print("bravo!\n")
print("valeur de la variable 0",%0,"\n")
print("toto".%0,"\n")
```

La dernière forme de l'instruction `print` permet d'obtenir les caractéristiques d'une primitive.

10.7.1 Format d'impression

Le format de l'impression peut être contrôlé par l'instruction `format` suivie d'une spécification utilisant la syntaxe du C. Par exemple

```
%0=1.23456
format "%4.1f"
print(%10,"\n")
```

donne comme résultat l'impression de la chaîne "1.2".

10.8 Style des traits

La commande `set style` permet de définir le style des lignes qui sera utilisé pour créer les primitives apparaissant dans la suite du programme. La syntaxe en est:

```
set style [thin][thick][dashed][thin dashed][thick dashed]
set style [vect][double_vect]
```

10.9 Entrée de valeurs au clavier

Il est possible de demander la valeur d'une variable ou d'un vecteur avec l'instruction `input_value`:

```
%10=input_value
%10=input_value
```

Le programme attend la valeur dans la fenêtre de texte.

10.10 Entrée de points à la souris

Au lieu d'utiliser des variables décrites de manière rigide dans le programme (ou définies comme argument au lancement de `xjpdraw`) on peut définir des points à la souris et récupérer leurs coordonnées dans le programme. Pour cela on utilise l'instruction `input_mouse` suivi du nombre de points attendus. A la première interprétation du programme les ordres sont exécutés jusqu'à cette instruction. Après cette instruction `xjpdraw` émet un message dans la fenêtre d'information en indiquant qu'il attend des points. On doit alors passer en mode "Sélection" et cliquer avec le bouton de gauche pour indiquer un point. Si le point n'est pas satisfaisant il peut être annulé en cliquant avec le bouton du milieu (ainsi que les points précédents d'ailleurs). Si le programme n'est pas celui qui convient une action avec le bouton de droite permet de sortir de l'utilisation du programme.

Les coordonnées des points entrés sont placées dans les vecteurs `%v0`,`%v1` et ainsi de suite. Lorsque le nombre de points requis a été introduit le programme est à nouveau exécuté. Il faut noter que les primitives qui ont été crées préalablement à la rencontre de cette instruction vont être répliqués.

10.11 Emission de message vers la console

Il est possible d'émêttre des messages sur la 2eme fenêtre d'information de `xjpdraw` par la commande `info`:

```
info "Bravo le dessin est fini"
```

Si vous utilisez l'instruction `input_mouse` il est nécessaire de placer l'instruction `info` avant cette instruction. Il est de bonne pratique d'émêttre un message d'information avant toute demande d'entrée de points à la souris, au moins pour indiquer le nombre de points à entrer.

10.12 Création de primitives

Les primitives sont numérotées dans l'ordre de leur création mais on peut leur affecter un autre numéro. Un numéro de primitive est défini par un # suivie d'un entier. Dans tout ce qui suit on peut toujours affecter un numéro de primitive à la primitive crée.

10.12.1 Texte

Les textes sont crées par des ordres du type:

```
text at [vecteur] [chaine] size [valeur]
[numero primitive]=text at [vecteur] [chaine] size [valeur]
```

où `valeur` est la taille des caractères (8, 10, 12, 14, 16, 18, 24) et `vecteur` la position du coin inférieur gauche du texte.

10.12.2 Cercles, Ellipses, Arcs de cercle

Ces éléments peuvent être crées à l'aide des instructions suivantes:

```
circle center [vecteur] radius [valeur]
circle toward [vecteur] [vecteur] [vecteur]
ellipsis center [vecteur] radius [valeur1] , [valeur2]
```

Dans le premier cas on donne la position du centre du cercle dans `vecteur` et son rayon dans `valeur`. Dans le second on définit les coordonnées de trois points sur son périmètre. Le troisième cas permet de définir une ellipse en donnant son centre (`vecteur`), sa largeur (`valeur1`) et sa hauteur (`valeur2`).

On peut récupérer le rayon d'un cercle dans une variable par:

```
[variable]=radius [numero primitive]
```

Des arcs de cercle peuvent être crée avec l'instruction:

```
arc center [vecteur] radius [valeur] from [valeur] to [valeur]
```

On définit dans l'ordre son centre, le rayon du cercle associé ainsi que les angles de départ et de fin de l'arc, en degré.

Une autre possibilité est d'utiliser la syntaxe

```
arc center %v0 at %v1 to %v2
```

qui crée un arc centré à `%v0` admettant comme points extrême les points dont les coordonnées sont dans `%v1, %v2`.

10.12.3 Rectangle

On peut crée des rectangles avec les instructions:

```
rectangle from [vecteur] to [vecteur]
rectangle from [vecteur] length [vecteur]
```

Dans le premier cas on définit la position de deux points opposés du rectangle et dans le second la position du coin supérieur gauche puis la largeur et la hauteur.

10.12.4 Ligne

Les syntaxes possibles sont:

1. `line from [vecteur] to [vecteur]`: on définit la position des points de départ et d'arrivée
2. `line from [vecteur], vector [vecteur], length [valeur]`: on donne la position du point de départ, le vecteur directeur et la longueur de la ligne
3. `line from [vecteur] [left/right] perpendicular to [numero primitive], length [valeur]`: on donne le point de départ et la ligne sera perpendiculaire à la primitive donnée, du côté indiqué par `left` ou `right`, de la longueur indiqué.
4. `line from [vecteur] parallel to [numero primitive], length [valeur]`: la ligne sera parallèle à la primitive donnée, de la longueur indiquée.

10.12.5 Polygone

Il est possible de créer un polygone en indiquant la position des sommets dans des vecteurs numérotées successivement (par exemple de `%v0` à `%v10`) puis utilisez l'instruction:

```
polygon from %v0 to %v10
```

10.12.6 Courbes

10.12.6.1 Spline

Il est possible de créer une courbe spline passant par des points de contrôle calculés. Pour cela placer la position de vos points de contrôle dans des vecteurs numérotées successivement (par exemple de `%v0` à `%v10`) puis utilisez l'instruction:

```
spline from %v0 to %v10
```

Une autre manière d'obtenir une courbe fermée "douce" consiste à utiliser les sommets d'un polygone existant comme points de contrôle:

```
#1=polygon from %v0 to %v10
spline #1
```

10.12.6.2 Formule

Il est possible de tracer une courbe du type $y = f(x)$ par une instruction du type:

```
Formule f(x), x between x1,x2 , step [valeur] , start [vecteur]
```

comme par exemple:

```
%v0[0]= 2
%v0[1]= 10
Formule sin(x), x between 1,10 , step 0.1 , start %v0
quit
```

qui va permettre de tracer la courbe $\sin(x)$ pour x variant de 1 à 10 par pas de 0.1, le premier point étant tracé aux coordonnées (2,10).

10.13 Primitive 3D

Il est possible de définir des primitives 3D avec le langage.

10.13.1 Définir la vue

Avant de définir des primitives 3D il est nécessaire de définir la vue utilisée. Pour cela il faut indiquer avec l'instruction `view` une sphère englobant la vue 3D et le cercle qui lui correspond en 2D. Par exemple:

```
view ( 0 0 0 7) ( 6 10 10)
```

indique que la sphère 3D est centrée en (0,0,0) et a un rayon de 7 et correspond à l'écran a un cercle centré en 6cm, 10cm de rayon 10cm.

10.13.2 Cercle 3D

On peut définir un cercle en 3D par :

```
cercle3d x1 y1 z1 radius r1 (psi1 teta1)
```

où `x1, y1, z1` sont les coordonnées du centre du cercle, `r1` son rayon et `psi1, teta1` les angles d'Euler en degré permettant de définir la normale au plan contenant le cercle (`psi1` définit la première rotation autour de z, `teta1` la rotation autour du nouveau x).

Par exemple:

```
cercle3d %82 0 0 radius %81 (0 90)
```

définit un cercle qui est dans le plan $x - z$.

10.13.3 Sphère 3D

On définit une sphère 3D par l'ordre:

```
sphere x1 y1 z1 radius r1 style [0,1,2]
```

où la valeur suivant `style` indique le type d'affichage de la sphère selon le codage: 0= sphère ombrée, 1=sphère avec facette, 2=sphère dessinée symboliquement. Les paramètres `x1, y1, z1` sont les coordonnées du centre de la sphère et `r1` son rayon.

10.13.4 Parallélépipède 3D

On peut définir un parallélépipède 3D par:

```
cube expr expr expr radius expr expr expr style expr
```

où `expr` sont des expressions élémentaires. Les trois premières définissent la position du centre du cube, les trois suivantes la hauteur, longueur, largeur du cube et le paramètre suivant `style` indiquant le mode de représentation du cube selon le même codage que pour les sphères.

10.13.5 Point 3D

Pour tracer des courbes en 3D on peut utiliser l'ordre `point3d` qui permet de récupérer dans un vecteur la position de la projection d'un point en 3D. La syntaxe est:

```
[vecteur]=point3d ( x1 y1 z1)
```

où `x1, y1, z1` sont les coordonnées 3D du point.

10.14 Texture de primitives

On peut texturer une primitive avec l'ordre:

```
texture [numero primitive] type y v1 x v2 at v3
```

où `type` vaut 1 pour les textures à lignes verticales, 2 à lignes horizontales, 3 à lignes verticales et horizontales, 4 à lignes penchées et 5 pour les textures à niveau de gris. Pour les textures de type 1 à 4 `v1` est la distance entre les lignes horizontales, `v2` la distance entre les lignes verticales. Pour les textures de type 5 `v3` est le niveau de gris, de 1 à 100, allant du noir au blanc total.

10.15 Manipulation des primitives

10.15.1 Destruction de primitives

Vous pouvez détruire une primitive par:

```
destroy [numero primitive]
```

10.15.2 Association, dissociation

On peut associer ou dissocier toute les primitives par les instructions `associate-all`, `dissociate-all`. Il est possible d'associer ou de dissocier les primitives individuellement par: `associate [numero primitive]` et `dissociate [numero primitive]`

Il est aussi possible manipuler des primitives qui ont été associées avant le lancement du programme en leur affectant un numéro de primitive. Pour cela on écrit:

```
#[nombre]=&[nombre1]
```

où `nombre1` correspond à l'ordre dans lequel ont été associées les primitives, commencant à 0 (la première primitive associée à le numéro 0).

10.15.3 Translation, rotation

Il est possible de translater une primitive par:

```
translate #[numero] [vecteur]
```

où `vecteur` définit le vecteur de translation. On peut aussi faire tourner une primitive par:

```
rotate #[numero] around [vecteur] , [valeur]
```

où `vecteur` définit le centre de rotation et `valeur` l'angle de rotation en radian. Ces transformations s'appliquent sur les primitives associées.

10.15.4 Compilation et taille des textes

L'instruction `compile_all_text` permet de demander la compilation des textes pour obtenir leur encombrement exact. Depuis la version 4.2 un processus de compilation tourne en tâche de fond en permanence ce qui permet de réduire très nettement le temps de calcul de la boite englobante, à condition toutefois de ne pas enchaîner très rapidement l'introduction de textes.

10.15.5 Copier une primitive

On peut copier une primitive par:

```
#[numero]= copy #[numero1]
```

qui copie la primitive `numero1` dans la primitive `numero`. Une autre possibilité est de copier avec homothétie par:

```
#[numero]= copyh #[numero1] [valeur]
```

qui copie avec un facteur d'homothétie donné par `valeur`.

10.15.6 Miroirs

Il est possible de faire des miroirs par les ordres:

```
lower-mirror #[numero]
upper-mirror #[numero]
right-mirror #[numero]
left-mirror #[numero]
```

10.15.7 Centrer

Il est possible de centrer l'ensemble des primitives, ou seulement les primitives associées par `Center`, `Center-associate`

10.16 Programme paramétré

Il est possible de paramétrer les programmes en indiquant au lancement des arguments qui sont soit des valeurs numériques ou des caractères. La syntaxe est:

`xjpdraw -F [nom du fichier programme][[argument0,argument1]]`

Par exemple:

`xjpdraw -F formule.f[0,2,toto]`

10.16.1 Déclaration des arguments

Pour indiquer le nombre d'argument on peut placer en tête du programme l'instruction:

`prog([nombre d'argument])`

Ce n'est pas obligatoire mais si cette instruction est présente le programme vérifiera le nombre des arguments fournis et refusera une instruction dans le programme où un argument de numéro incorrect est utilisé.

10.16.2 Récupération des arguments

Les arguments sont numérotés dans leur ordre d'apparition. Leur valeur est contenue dans un nom qui commence par $ suivie d'un entier, commencant à 0.

10.16.3 Unitlength

Vous pouvez ajuster la variable `unitlength` par:

`set unitlength [nombre]`

10.16.4 Commentaires

Tout caractère compris entre `/*` et`*/` est considéré comme un commentaire.

10.16.5 Sauver et charger un fichier

Pour charger un fichier `.macdraw` utilisez:

`load [chaine de caractere]`

Pour sauver vous pouvez utiliser:

`save [chaine de caractere]`

qui sauve le fichier Postscript à part ou bien:

`save_included [chaine de caractere]`

qui sauvera le Postscript dans le fichier LaTeX. Par exemple:

```
save "toto.ltex"
save $0
load "tata"
```

Ces instructions permettent de modifier de manière très souple des fichiers `xjpdraw`. Considérons par exemple le programme `zoom.f` suivant:

```
load $0
set unitlength $2
save $1
quit
```

et que l'on lance avec

`xjpdraw -FI zoom.f[toto,toto.ltex,0.5]`

Le fichier `toto.macdraw` est alors chargé, `unitlength` est mis à 0.5 et le fichier est sauvé. Il est alors deux fois plus petit que l'original. L'exécution est ici lancée en invisible et peut donc être effectuée à partir d'un simple terminal.

10.16.6 Fin de programme

Pour terminer un programme utilisez:

```
quit
```

10.17 Lire une fonction utilisateur

Par programme on peut aussi charger une fonction utilisateur en utilisant la même syntaxe que si on la chargeait au moment du lancement de `xjpdraw`. Ainsi:

```
load-function "-X 10 -Y 10 -F toto"
```

va charger la fonction `toto` dans un rectangle de dimension 10,10. Les autres syntaxes possibles sont:

```
load-function "-X [expr] -Y [expr] -F [chaine]"
load-function "-X [expr] -Y [expr] -F [chaine] -O [expr]
[expr] -F [chaine]"
```

comme par exemple:

```
load-function "-X %12 -Y %12 -O %10 %11 -F toto".%0
```

Il est possible aussi de tester l'existence d'une fonction utilisateur (voir `exist`).

On peut aussi ajuster les paramètres du quadrillage qui va être placé sur la fonction par:

```
grid x = [expr]
grid y = [expr]
format x [texte]
format y [texte]
```

Les deux premières instructions permettent d'ajuster les distances en cm entre les traits des échelles et les deux dernières de modifier le format d'impression des valeurs sur les échelles.

Le tracé du quadrillage est obtenu par l'instruction `put-echelle`. On peut aussi demander un quadrillage court ou long par les instructions `echelle-courte` ou `echelle-longue`.

Si l'on veut charger plusieurs fichiers fonctions sans modifier l'échelle utilisée on peut employer l'instruction `no_scale_change`. Si l'instruction est répétée on revient à la mise à l'échelle automatique.

Les expressions `function_max_x,function_min_x` contiennent les valeurs maximum et minimum en x d'une fonction chargée. Il en est de même pour les expressions `function_max_y,function_min_y` pour les valeurs en y.

10.18 Divers

10.18.1 Redémarrage

L'équivalent de la commande `xjpdraw Redémarrer tout` est obtenu par l'instruction `new`.

10.18.2 Distance entre deux points

Pour calculer la distance entre deux points:

```
[variable]=distance from [vecteur] to [vecteur]
```

10.18.3 Centre d'une primitive

Pour obtenir les coordonnées du centre d'une primitive utilisez:

```
[vecteur]=center [numero primitive]
```

10.18.4 Intersection de deux segments

Vous pouvez calculer les coordonnées du point d'intersection de deux segments avec:

```
[vecteur]=intersection [numero primitive] [numero primitive]
```

Si il n'y a pas d'intersection les coordonnées du vecteur seront mise à -1.

10.18.5 Coordonnées de départ et de fin d'un segment

Les coordonnées des extrémités d'une ligne peuvent être obtenues par:

```
[vecteur]=start [numero primitive]
[vecteur]=end [numero primitive]
```

10.19 Existence d'un fichier

On peut tester si un fichier existe par la commande suivante :

```
[variable]= exist [chaine]
```

La variable vaut 1 si le fichier existe, 0 sinon. Par exemple

```
%10=exist "toto"
%11=exist "toto".%2
```

10.20 Ordre d'écriture des fichiers Postscript, LaTeX

Par défaut dans le fichier LaTeX crée par `xjpdraw` on introduit dans l'ordre le fichier Postscript puis les textes LaTeX. Il est possible d'inverser cette ordre d'écriture en utilisant le mot-clé `ps_after_latex`. L'ordre par défaut sera rétabli si l'on utilise `ps_before_latex`.

10.21 Exemples de programme

10.21.1 Exemple d'interaction avec la souris

```
/*----------------------------------------
Programme de construction d'actionneur lineaire en
Jpdraw.

On donne 2 points pour definir dans l'ordre l'attache de l'actionneur sur le corps
puis sur la tige
----------------------------------------*/

/* information */

info "Construction d'actionneur lineaire. En mode selection Droite=abort"
/*les point initiaux */
input_mouse 2


%1=1.5

/* diametre actionneur, piston */
%2=%1/8
%12=%1/10

%v2[0]= %v1[0]-%v0[0]
```

```
%v2[1]= %v1[1]-%v0[1]
%5= sqrt(%v2[0]*%v2[0]+%v2[1]*%v2[1])
%v20[0]=%v2[0]/%5
%v20[1]=%v2[1]/%5

%v5[0]=%v2[1]
%v5[1]=%v2[0]

%9=atan2(%v5)
%6=%9*180/pi

/* le corps de l'actionneur */

%v10[0]=%v0[0]+%1*%v20[0]-%2*cos(%9)
%v10[1]=%v0[1]+%1*%v20[1]+%2*sin(%9)

%v11[0]=%v0[0]-%2*cos(%9)
%v11[1]=%v0[1]+%2*sin(%9)

%v12[0]=%v0[0]+%2*cos(%9)
%v12[1]=%v0[1]-%2*sin(%9)

%v13[0]=%v0[0]+%1*%v20[0]+%2*cos(%9)
%v13[1]=%v0[1]+%1*%v20[1]-%2*sin(%9)

line from %v10 to %v11
line from %v11 to %v12
line from %v12 to %v13



%v14[0]=(%v0[0]+%v0[0]+%1*%v20[0])/2
%v14[1]=(%v0[1]+%v0[1]+%1*%v20[1])/2

/* la tige de l'actionneur */


%v15[0]=%v14[0]-%12*cos(%9)
%v15[1]=%v14[1]+%12*sin(%9)

%v16[0]=%v14[0]+%12*cos(%9)
%v16[1]=%v14[1]-%12*sin(%9)

line from %v15 to %v16
line from %v14 to %v1

quit
```

Ce programme a été utilisé pour créer le dessin de la figure 10.1.

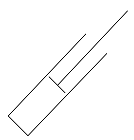


FIG. 10.1 -

10.21.2 Exemple d'utilisation du `for`

```
/*------------------------
Trace d'un cercle avec des rayons qui parte du centre
sur lesquels on place des points a intervalles regulier
-------------------------*/
for %0 from 0; (%0 < 360); step 10 bloc
       %10=%0*pi/180
       %1=5*cos(%10)
       %2=5*sin(%10)
       %v0=(6,8)
       %v1=(6+%1,8+%2)
       line from %v0 to %v1
       for %20 from 0; (%20 < 5); step 0.5 bloc
              %11=%20*cos(%10)
              %12=%20*sin(%10)
              %v10=(6+%11,8+%12)
              circle center %v10 radius 0.02
       end_bloc
end_bloc
circle center %v0 radius 5
quit
```

La figure 10.2 en donne un exemple.

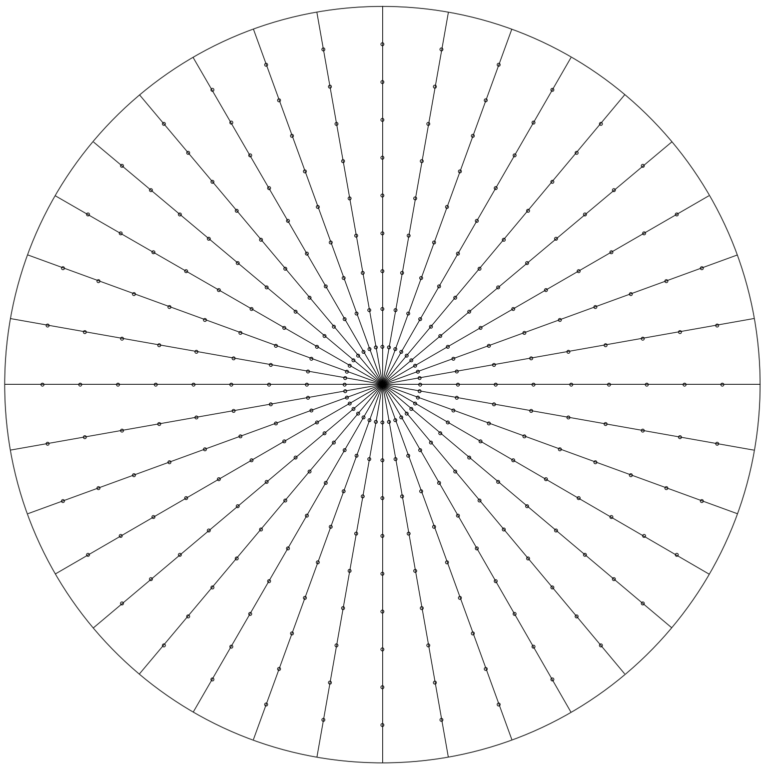


Fig. 10.2 -

10.21.3 Exemple d'utilisation de `Formule`

```
/* ---------------------
Trace d'une ligne sinusoidale entre deux
```

```
points donnes a la souris
----------------------- */

info "Wiggly line,ligne sinusoidale de A a B,En mode selection Droite=abort"

input_mouse 2

%1=distance from %v0 to %v1
%2=atan((%v1[1]-%v0[1])/(%v1[0]-%v0[0]))

%3=0.2 /* k1 */
%4=10 /*k2*/
%6=pi/(2*%4)+%1/2 /*u2*/
%5=%6-%1

#0=Formule %3*sin(%4*x), x between %5,%6 , step 0.05 , start %v0

rotate #0 around %v0 , %2

quit
```

Un exemple d'utilisation de ce programme est présenté en figure 10.3.

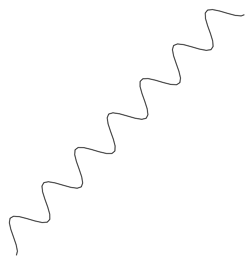


FIG. 10.3 -

10.21.4 Exemples complexes

```
/*----------------------------------------
Programme de construction de faisceau de cercle par
Jpdraw a partir de deux points entres a la souris.
Convention: toute valeur en cm
%[0-9]              : variable flottante
%v[0-9]              : vecteur de dimension 2, composante accessible par
              %v[0-9][0],%v[0-9][1]
#[0-9]              : primitive

On donne ici a la souris deux points de coordonnees %v0, %v1.
On trace le segment S1 passant par ces deux points. On calcule le milieu M
de ce segment (%v2) et on trace la mediatrice S2 de ce segment.
On definit deux points P1,P2 sur le segment S1 et un point P3 sur S2.
On trace le cercle C passant par P1, P2, P3. On choisit deux points Q1, Q2
sur ce cercle et on calcule les cercles normal a C passant par Q1, Q2 dont les
centres sont sur S1. On trace ensuite les tangentes a ces cercles qui passent par
M. Par construction ces deux tangentes ont meme longueur. On affiche les longueurs
----------------------------------------*/

/* information */
```

```
info "Construction de faisceau de cercle. En mode selection Droite=abort"
/*les point initiaux */
input_mouse 2

/*on suppose que l'axe x est S1 : on memorise l'angle de rotation */
%80=atan((%v1[1]-%v0[1])/(%v1[0]-%v0[0]))
/* le milieu M de v0, v1 */
%v2[0]=(%v0[0]+%v1[0])/2
%v2[1]=(%v0[1]+%v1[1])/2

/*le segment S1 */
#1=line from %v0 to %v1

/*S2=mediatrice a S1 */
#2=line from %v2 left perpendicular to #1 , length 3
#3=line from %v2 right perpendicular to #1 , length 3

/*v3: un point sur S1 */
%v3[0]=(%v2[0]+%v1[0])/2
%v3[1]=(%v2[1]+%v1[1])/2

/*v4: un deuxieme point sur S1 */
%v4[0]=(%v2[0]+%v0[0])/2
%v4[1]=(%v2[1]+%v0[1])/2

/*v6: un point sur S2 */
%v5=end #2
%v6[0]=(%v2[0]+%v5[0])/2
%v6[1]=(%v2[1]+%v5[1])/2

/*le cercle C passant par v3,v4,v6 */

#4=circle toward %v3 %v4 %v6

/* %v7 centre du cercle C, %8 son rayon */
%v7= center #4
%8=radius #4

/*#5,#6 deux lignes issus du centre de C */

#5=line from %v7 , vector (1,1) , length %8
#6=line from %v7 , vector (1,-1) , length %8

/*#7,#8 deux lignes perpendiculaires aux precedentes */

%v9=end #5
%v10= end #6
#7=line from %v9 , vector (1,-1) , length 7
#8=line from %v10 , vector (1,1) , length 7

/* intersection de #7,#8 avec S1, les futurs centres des deux cercles
normaux a C */

%v11=intersection #1 #7
%v12=intersection #1 #8

/* le rayon de ces cercles */

%10=distance from %v9 to %v11
%11=distance from %v10 to %v12
```

```
/*construction des cercles */

circle center %v11 radius %10
circle center %v12 radius %11

/*construction des tangentes a ces cercles passant par v2 */

%13=distance from %v2 to %v11
%14=sqrt(%13*%13-%10*%10)
%15=atan(%10/%14)
%v30=(cos(%15+%80),sin(%15+%80))

set style thick
#9=line from %v2 , vector %v30 , length %14

%23=distance from %v2 to %v12
%24=sqrt(%23*%23-%11*%11)
%25=atan(%11/%24)
%v31=(cos(%25+%80),sin(%25+%80))

set style thick
#10=line from %v2 , vector %v31 , length %24

/* print(%14,%24) */

/* destruction de primitive utilisee pour la construction */

destroy #5
destroy #6
destroy #7
destroy #8

/*ecriture des differents textes */

%v40[0]=%v2[0]-0.3
%v40[1]=%v2[1]+0.3
text at %v40 "A" size 10

%v41=end #9
%v42=end #10

%v43[0]=%v41[0]
%v43[1]=%v41[1]

%v44[0]=%v42[0]
%v44[1]=%v42[1]
text at %v43 "B1" size 10
text at %v44 "B2" size 10

%v50=end #2
%v60=end #3


text at %v50 "Distance AB1",%14 size 10

%v51[0]=%v50[0]
%v51[1]=%v50[1]-0.5
text at %v51 "Distance AB2",%24 size 10
```

```
%v61[0]=%v60[0]-3
%v61[1]=%v60[1]-0.5
text at %v61 "Construction automatique de faisceau de cercle par JpDraw"
size 10
set style thin
quit
```

Un exemple d'utilisation de ce programme est donné en figure 10.4.

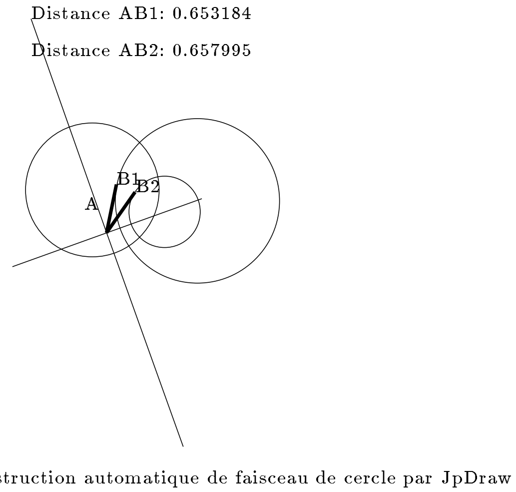


FIG. 10.4 -

10.21.5 Exemples en 3D

```
/*----------------------------
Decoupage d'un cube 3D en petit sous-cube
-----------------------------*/
view ( 0 0 0 15) ( 6 10 5)

%20=5

%10= -7.5+%20/2
%11= 7.5-%20
for %0 from %10 ; (%0 <=%11) ; step %20 bloc
       for %1 from %10 ; (%1 <= %11) ; step %20 bloc
              for %2 from %10 ; (%2 <= %11) ; step %20 bloc
                     cube %0 %1 %2 radius %20 %20 %20 style 2
              end_bloc
       end_bloc
end_bloc
quit
```

On obtient alors le dessin de la figure 10.5.

10.21.6 Surface de Connoly

```
/*--------------------------
Programme pour surface de Connoly: surface generee par la
trace des spheres qui roulent sur deux spheres fixes.
Les spheres sur lesquelles on roule sont
sur l'axe x par definition, symetrique par rapport a
```

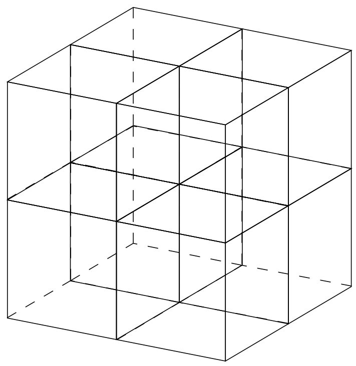


FIG. 10.5 -

```
l'origine (coordonnee %0, rayon %10,%11)
sphere roulante: rayon=%20, centre %21,%22,%23
On trace %70 sphere qui roule
--------------------------*/
view ( 0 0 0 7) ( 6 10 10)
alias_variable "r1" 10
alias_variable "r2" 11
alias_variable "x1" 0
alias_variable "x2" 1
alias_variable "r" 20
alias_variable "D" 40
alias_variable "d1" 50
alias_variable "d2" 51
alias_variable "R" 61
alias_variable "xc" 21
alias_variable "yc" 22
alias_variable "zc" 23
{x1}=1.5
{x2}= -{x1}
{r1}=1
{r2}=1
/* les spheres sur lesquells on roule */

set style thick
sphere {x1} 0 0 radius {r1} style 2
sphere {x2} 0 0 radius {r2} style 2
set style thin


{r}=1.5

/* A1, A2 */
%30=({r1}+{r})**2
%31=({r2}+{r})**2

/* D */

{D}=2*{x1}

/* d1, d2, D=d1+d2 */

{d2}=({D}*{D}-%30+%31)/(2*{D})
{d1}={D}-{d2}
```

```
{xc}={x2}+{d1}

/* R^2 =%60*/

%60= %30-{d1}*{d1}

if ( %60 < 0 ) bloc
       print("pas de solution\n")
       quit
       end_bloc
/* R = %61 */

{R}=sqrt(%60)


/* on trace %70 sphere qui roule */

%70=5

for %71 from 1; (%71 < %70) ; step 1 bloc
       {yc}={R}*cos(2*pi*%71/%70)
       {zc}={R}*sin(2*pi*%71/%70)
       sphere {xc} {yc} {zc} radius {r} style 2
end_bloc

%v0[0]={d1}
%v0[1]={R}

%80=atan2(%v0)

%81={r1}*sin(%80)
%82={r1}*cos(%80)

%82={x1}-%82
cercle3d %82 0 0 radius %81 (90 90)

if ( %60 > {r} ) bloc
       %v1[0]={d1}
       %v1[1]={R}-{r}
       %85=atan2(%v1)
       end_bloc
if (%60 <= {r})bloc
       %85=0
       end_bloc

for %40 from %85; (%40 < %80) ; step 0.1 bloc
       %41={d1}*cos(%40)+{R}*sin(%40)
       %41= -%41
       %42={d1}*{d1}+{R}*{R}-{r}*{r}
       %45= -%41-sqrt(%41*%41-%42)
       %81=%45*sin(%40)
       %82=%45*cos(%40)
       %82={x1}-%82
       cercle3d %82 0 0 radius %81 (90 90)
       end_bloc


%v0[0]={d2}
%v0[1]={R}
```

```
%80=atan2(%v0)

%81={r2}*sin(%80)
%82={r2}*cos(%80)

%82={x2}+%82
cercle3d %82 0 0 radius %81 (90 90)


if ( %60 > {r} ) bloc
       %v1[0]={d2}
       %v1[1]={R}-{r}
       %85=atan2(%v1)
       end_bloc
if (%60 <= {r})bloc
       %85=0
       end_bloc

for %40 from %85; (%40 < %80) ; step 0.1 bloc
       %41={d2}*cos(%40)+{R}*sin(%40)
       %41= -%41
       %42={d2}*{d2}+{R}*{R}-{r}*{r}
       %45= -%41-sqrt(%41*%41-%42)
       %81=%45*sin(%40)
       %82=%45*cos(%40)
       %82={x2}+%82
       cercle3d %82 0 0 radius %81 (90 90)
       end_bloc
```

Une surface de Connoly est tracée en figure 10.6.

10.21.7 Exemples de chargement de fonction utilisateur

```
/* charge des fichiers fonction de nom toto suivie d'un nombre
On teste l'existence de ces fichiers et on les met dans des carres
de dimension fixe avec un grillage */

/* origine et dimension du cadre */
%10=0.8
%11=0.5
%12=4
%13=0.8
%14=1

%20=0

/* format d'ecriture des noms */
format "%5.0f"

for %0 from 10; (%0 < 350);step 59 bloc
       %1=exist "toto".%0
       if (%1==1)bloc
              load-function "-X %12 -Y %12 -O %10 %11 -F toto".%0
              echelle-courte
              grid x = 1
              grid y = 1
              format x "3.1"
              format y "3.1"
              put-echelle
```

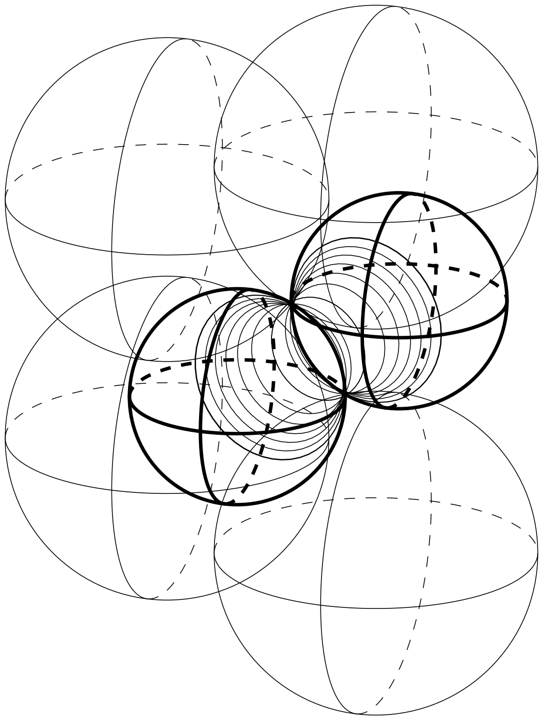


FIG. 10.6 - *Surface de Connoly*

```
            %v0[0]=%10+1
            %v0[1]=14.5-%11-1
            text at %v0 %0 size 10
            %20=%20+1
            %10=0.8+%20*(%12+%13)
            if (%20 > 2) bloc
                  %20=0
                  %10=0.8
                  %11=%11+(%12+%14)
            end_bloc
      end_bloc
end_bloc

quit
```

Chapitre 11

Arguments

Dans ce chapitre on va indiquer quelles sont les arguments que peut prendre `xjpdraw` au moment du lancement du programme et l'on va donner un résumé des touches fonctions.

11.1 Arguments

On peut indiquer plusieurs type d'argument à `xjpdraw`.

11.1.1 Un fichier .macdraw

Le 1er est un nom de fichier `.macdraw` que l'on va charger directement (l'extension `.macdraw` n'est pas indispensable si il y a une seule extension).

`xjpdraw toto` si il existe un fichier toto.macdraw mais `xjpdraw toto.ps.macdraw` pour le fichier toto.ps.macdraw

11.1.2 Un fichier fonction

Le 2eme type d'argument que l'on peut donner revient à entrer directement une fonction utilisateur (voir le chapitre "Fonction utilisateur"). La syntaxe est alors:

```
xjpdraw -X x1 -Y y1 -F nom [-E xmin xmax ymin ymax][-O x2 y2]
```

où `x1` représente la largeur en cm de la boite où l'on va tracer la fonction, `y1` sa hauteur en cm. `nom` est le nom du fichier fonction. Si en place de `-F` on met `-Fp` on ne tracera pas le cadre englobant (il sera toutefois présent dans le presse-papier). Les arguments optionnels sont:

- `-E` permet de tracer le dessin avec une échelle fixée. Les arguments représentent les coordonnées minimum et maximum de la fonction dans la boite. Les arguments de `-E` sont en unité de la fonction.
- `-O` permet de fixer la position du coin supérieur gauche de la boite (par défaut il est placé à 2cm en x et y du coin supérieur gauche du canevas). Les arguments de `-O` sont en cm.

11.1.3 Un programme

Pour lancer un programme (voir le chapitre "Programmation") utiliser:

```
xjpdraw -F [nom de programme]
```

Vous pouvez aussi donner des arguments à un programme:

```
xjpdraw -F [program file name][[argument0,argument1]]
```

comme dans:

```
xjpdraw -F formule.f[0,2,toto]
```

11.1.4 Mémoire

Nous avons vu que par défaut le nombre de primitives disponibles dans `xjpdraw` est de 1500. Pour certaines utilisations ce nombre peut toutefois s'avérer insuffisant. On peut modifier au lancement du programme ce nombre en lancant `xjpdraw` avec l'option `-M` suivie du nombre de primitives à initialiser. Ainsi

```
xjpdraw -M 2000
```

permet l'utilisation de 2000 primitives. L'allocation de mémoire est automatique lorsque l'on charge un fichier `.macdraw` avec plus de 1500 primitives. Par contre ce n'est pas le cas si l'on utilise l'album.

11.1.5 Vérification d'un configurateur

Si comme argument on utilise `-check`, `xjpdraw` vérifiera le configurateur et émettra un diagnostic sur la validité du configurateur (voir page 112).

11.1.6 Mode transparent

Pour créer un document du type transparents on utilisera l'option `-slide`.

11.2 Résumé de l'utilisation des touches fonctions et des raccourcis

Pour utiliser un raccourci la souris doit être dans le canevas.

touche	mode	action
L2	rectangle	force le tracé d'un carré
L2	ellipse	tracé de cercles
L3	polygone	polygone en forme de tête de vecteur
L3	sélection	sélection de la 2eme primitive recouvrante
L4	rectangle	revient en mode rectangle
L4	polygone	détruit dernière arête
L4	ellipse	tracé d'ellipse
L5	polygone	annule la création du polygone
L6	sélection	permet ou empêche de déplacer avec la souris
L6	cercle	supprime l'arc de cercle de longueur maximal
L8	rectangle	associe les primitives internes du rectangle
L8	ligne	force la pente de la ligne a une pente LaTeX standard
L8	cercle	supprime l'arc de cercle de longueur minimale
L9	sélection	grand déplacement d'une primitive
L9	polygone	force le tracé d'un polygone régulier
L9	rotation	pour indiquer un centre de rotation
L10	polygone	revient au mode polygone général
L10	sélection	changement de taille d'une primitive ou emacs
F1	-	sélectionne l'icône sélection
F2	-	sélectionne l'icône de texte
F3	-	sélectionne l'icône de ligne
F4	-	sélectionne l'icône de rectangle
F5	-	redessine tout
F6	-	sélectionne l'icône des ellipses
F7	-	sélectionne l'icône des polygones
F8	-	sélectionne l'icône du crayon
F9	-	sélectionne l'icône de rotation
R1	sélection	translation d'un pixel vers la gauche
R2	sélection	translation d'un pixel vers la droite
R4	sélection	translation d'un pixel vers le haut
R4	polygone	détruit dernière arête
R5	polygone	annule la création du polygone
R5	sélection	translation d'un pixel vers le bas
R6	-	raccourci pour copier
R7	-	raccourci pour coller
R13	-	raccourci pour couper

Si on est en mode "Sélection" on dispose d'un certain nombre de raccourci pour les commandes.

touche	action
a	affichage du presse-papier
A	associer tout
c	centre le dessin
d	dissocier la primitive sélectionnée
D	dissocier tout
f	ajoute une flêche au polygone sélectionné
h	affiche l'aide en ligne
m	masquage du presse papier
n	ligne perpendiculaire à la sélectionnée
p	passe la sélectionnée au premier plan
q	affiche ou masque le quadrillage
r	affiche ou masque les règles en centimètres
s	passe la sélectionnée au second plan
t	centre un texte associé dans la primitive associée
w	sauve le dessin
[0-9]P	exécution du programme [0-9]

Si l'on n'est pas en mode "Sélection" certains raccourcis sont aussi disponibles. Certains raccourcis sont d'ailleurs indiqués dans les menus. Les touches CTRL ont les fonctions suivantes (les raccourcis indiqués consiste à appuyer sur la touche `CTRL`, puis à appuyer sur le caractère suivant le tiret):

touche	action
CTRL-a	associer tout
CTRL-b	met/enléve une bordure
CTRL-c	centre le dessin
CTRL-d	dissocier tout
CTRL-e	compile et lance le previewer
CTRL-f	affiche le nom du fichier courant et le répertoire
CTRL-g	affiche la fenêtre des textures grises
CTRL-i	activation/désactivation de la réglure magnétique
CTRL-l	affiche les pointillés et passe en mode pointillé
CTRL-m	affichage du presse-papier
CTRL-n	passe au mode suivant de l'aimant
CTRL-o	taille du dessin
CTRL-p	passe au mode précédent de l'aimant
CTRL-q	affiche ou masque le quadrillage
CTRL-r	affiche ou masque les règles en centimètres
CTRL-s	passe la sélectionnée au second plan
CTRL-t	centre un texte associé dans la primitive associée
CTRL-u	donne le nombre de primitive dans le presse-papier
CTRL-w	sauve le fichier
CTRL-x	recompile pour le previewer
CTRL-y	affichage du presse-papier
CTRL-z	affiche l'aide en ligne
CTRL–	vider le presse papier

Chapitre 12

Installation

12.1 Les fichiers pour l'installation de `xjpdraw`

Les fichiers suivant sont optionnels pour l'installation de `xjpdraw`, version française:

```
header.latex
header_end.ltex
lplainext.tex
lplainext.psm
doc-en-ligne
```

Pour obtenir une version anglaise il convient de remplacer le dernier fichier par

```
doc-en-ligne-english
```

Il est bien entendu aussi nécessaire d'avoir l'exécutable `xjpdraw`.

Une bonne règle d'installation est de placer tous ces fichier dans un répertoire unique, quoique cela ne constitue pas une obligation.

Ces fichiers ont les rôles suivant:

- `header.latex`: ce fichier contient un préambule LaTeX standard. Il sera utilisé lors de l'usage des commandes `Vérification, Compile`. Il sera concaténé avec le fichier LaTeX crée par `xjpdraw` pour obtenir le fichier bidon.ltex, directement compilable par LaTeX. Si ce fichier n'existe pas `xjpdraw` mettra un préambule standard qui dépendra de la version de LaTeX telle que définie dans `LATEX_VERSION`,

  ```
  \documentstyle{article}
  begin{document}
  ```

  pour LaTeX 2.09 et

  ```
  \documentclass{article}
  begin{document}
  ```

  pour LaTeX2e.

- `header_end.ltex`: ce fichier contient simplement une fin de fichier LaTeX. Il sera concaténé avec les précédents. S'il n'est pas présent `xjpdraw` utilisera simplement `end{document}`.

- lplainext.tex,lplainext.psm: une série de macro LaTeX et Postscript permettant la rotation de texte dans un fichier LaTeX. L'usage de ces macros n'est pas obligatoire et elles peuvent être remplacées par d'autres, voir la commande `ROTATE_ORDER`.

- doc-en-ligne-english, doc-en-ligne : les fichiers contenant le manuel d'utilisation en ligne de `xjpdraw`. Ces fichiers ne sont pas obligatoires mais sans eux vous n'aurez pas d'aide en ligne.

L'installation de `xjpdraw` se fait donc selon les étapes suivantes:

1. placer tous les fichiers ci-dessus dans un répertoire unique (que l'on va appeler `toto`).
2. dans le fichier `header.latex` modifier le chemin d'accès au fichier `lplainext..` (à remplacer par `toto`)
3. dans le fichier `lplainext.tex` modifier le chemin d'accès au fichier `lplainext.psm` (à remplacer par `toto`)
4. dans le fichier `lplainext.tex` rechercher l'instruction

   ```
   \def\PostScriptCommandFile{lplainext.psm}
   ```

   et ajouter le chemin d'accès devant le nom du fichier `.psm`
5. mettez à jour le configurateur

12.2 Le fichier configurateur

Pour se configurer `xjpdraw` utilise un fichier, le `fichier configurateur` qu'il est absolument nécessaire de définir. Pour trouver ce fichier `xjpdraw` regarde la variable d'environnement `JPDRAW` qui doit indiquer la localisation et le nom du fichier configurateur.

Dans ce fichier on définit des commandes de la manière suivante: elles commencent toutes par un # suivi d'un mot-clé lui même suivi de paramètres. Les mots-clés sont:

```
BIB
BLINK
COMMAND_FILE
COMPILE_LATEX
CONTEXT
DVI->PS
EN_TETE_SLIDE
END_SLIDE
EXTERNE
FOLLOW_MOUSE
FONT
FONTMATH
FULL_MANUAL
GHOST
GHOSTVIEW
GHOST_UPDATE
GS
LANGUAGE
LATEX_COMMAND
LATEX_VERSION
LIB
LIB_EXTERNE
LOCAL_JPDRAW
MANUAL
PRINTING
PS
PSEXTERNE
ROTATE_ORDER
SELECTION_TYPE
SLIDE_HEIGHT
SLIDE_WIDTH
START_SLIDE
UNDO_DIR
UNDO_NUMBER
```

Le rôle de ces instructions est indiqué dans le tableau suivant:

Nom	Rôle
BIB	le chemin d'accès aux modules de programmation
BLINK	gère le clignotement des primitives
COMMAND_FILE	fichier de command LaTeX ajouté lors des compilations
COMPILE_LATEX	indique si l'on compile les textes
CONTEXT	charge un contexte
DVI->PS	programme qui permet de transformer les fichiers .dvi en fichier Postscript
EN_TETE_SLIDE	en-tête de document pour le mode transparent
END_SLIDE	fin d'instruction d'encapsulation de page en mode transparent
EXTERNE	pour gérer les programmes externes
FOLLOW_MOUSE	contrôle du mouvement des primitives en mode sélection
FONT	les fonts écran
FONTMATH	pour la visualisation à l'ecran des symboles mathématiques
FULL_MANUAL	fichier postscript contenant le manuel
GHOST	pour la visualisation du postscript d'un dessin
GHOST_UPDATE	utilisée pour mettre a jour le postscript
GHOSTVIEW	pour la visualisation du postscript du manuel
GS	pour la visualisation d'un dessin postscript externe
LANGUAGE	indique le langage utilisé
LATEX_COMMAND	la commande permettant de lancer LaTeX
LIB	chemin d'accès aux fichiers
LIB_EXTERNE	chemin d'accès aux programmes externes
LATEX_VERSION	la version de LaTeX utilisée
LOCAL_JPDRAW	le configurateur local
MANUAL	chemin d'accès au fichier de la doc en ligne
PRINTING	pour la compilation et l'impression d'un fichier
PS	permet d'indiquer une nouvelle syntaxe pour l'inclusion de Postscript
PSEXTERNE	pour l'incorporation de fichier postscript externe
ROTATE_ORDER	définit la rotation pour les textes
SELECTION_TYPE	indique comment est géré la sélection des primitives
SLIDE_HEIGHT	hauteur d'une page en mode transparent
SLIDE_WIDTH	largeur d'une page en mode transparent
START_SLIDE	début d'instruction d'encapsulation de page en mode transparent
UNDO_DIR	le répertoire où sont sauver les fichiers d'Annuler
UNDO_NUMBER	le seuil du nombre de primitive pour Annuler

Un second type de commande concerne les touches utilisées pour les raccourcis et certaines commandes spéciales de `xjpdraw`. Ces touches sont symboliquement appelées `R1,..,R15`, `L1,..,L10`, `F1,..,F9`. Dans le configurateur il sera possible d'affecter ces touches à n'importe quelle touche d'un clavier.

On peut placer des commentaires dans le configurateur en utilisant la syntaxe du langage C, c'est-à-dire en les placant entre `/*` et `*/`.

Il est aussi possible de lire un nouveau fichier configurateur en étant dans `xjpdraw` en utilisant la commande du menu `FICHIER`.

12.2.1 BIB

Lorsque l'on a écrit des modules de programmes d'utilité générale (voir le chapitre Programmation) avec l'extension est `.fi` et que l'on veut pouvoir les utiliser en permanence avec l'aide du bouton BIB de `xjpdraw` on les place dans un répertoire unique. Cette commande permet d'indiquer la localisation de ce répertoire comme, par exemple:

```
# BIB /u/merlet/Util_Jpdraw
```

12.2.2 BLINK

Cette commande permet de gérer le clignotement des primitives de `xjpdraw` lorsqu'elles sont sélectionnées. Le mot-clé `BLINK` est suivi d'un numéro qui indique le type de primitive concerné selon le code suivant:

0	texte
1	ligne
2	rectangle
3	cercle, ellipse
4	ovale
5	polygone

Pour les primitives de type 0 à 4 le numéro de la primitive est suivi d'un nombre indiquant le nombre d'extinction de la primitive lors de la sélection. Ainsi

```
# BLINK 0 4
```

indique que les textes clignotent 4 fois lors de la sélection.

Pour les polygones le numéro est suivie de deux nombres qui indiquent le nombre de clignotement pour les petits polygones (moins de 30points) et les grands.

```
# BLINK 5 20 2
```

indique que les petits polygones clignoteront 20 fois, les grands 2. Les valeurs par défaut sont:

0	6
1	20
2	11
3	4
4	4
5	11, 2

12.2.3 COMMAND_FILE

Cette commande est suivie d'un nom de fichier. Dans ce fichier on peut définir des nouvelles commandes LaTeX qui seront ajoutées au fichier LaTeX crée par `xjpdraw` lors de l'utilisation des commandes où l'on demande de créer des fichiers temporaires ne contenant que le dessin courant.

12.2.4 COMPILE_LATEX

Le but de cette commande est d'indiquer si l'on désire que les textes soient compilés pour avoir une estimation exacte de leur encombrement. Cette commande doit être suivie d'un chiffre, soit 0 (les textes ne seront pas compilés) ou 1. Ainsi si la commande `LATEX_COMMAND` est définie on pourra mettre:

```
# COMPILE_LATEX 1
```

12.2.5 CONTEXT

La commande `# CONTEXT` est suivie d'un nom de fichier, ce fichier décrivant un contexte (voir le menu `Fichier`). Ce contexte sera appliqué après le lancement du programme.

12.2.6 DVI->PS

Cette commande permet d'indiquer le type de programme qui est utilisé pour convertir les fichiers `.dvi` en fichier Postscript. Cette connaissance permet à `xjpdraw` de créer des fichiers postscripts compatibles avec ce programme. La syntaxe de cette commande est:

```
DVI->PS [nom de programme]
```

où [nom de programme] est soit dvi2ps, soit dvips, soit dvitops. L'usage de dvips est recommendé.

Dans tous les cas sera placée en tête du fichier Postscript une BoundingBox, en commentaire. Si l'on utilise dvi2ps le Postscript place le curseur à la position courante puis effectue le tracé. Dans le cas de dvips on suppose que l'instruction d'inclusion du Postscript a déjà placée le curseur à la position courante. On se contente donc de donner le dessin. Dans le cas de dvitops la gestion de la position courante est identique à celle de dvips mais on place en plus en tête du fichier l'instruction %! . Il est recommendé d'utiliser dvips.

12.2.7 EN_TETE_SLIDE

La chaîne qui suit cet ordre, jusqu'au prochain retour chariot sera utilisée comme en-tête de document du fichier bidon.ltex en mode transparent lors de l'utilisation de la commande Vérification.

12.2.8 END_SLIDE

La chaîne qui suit cet ordre sera incorporé après chaque page du document dans le fichier bidon.ltex en mode transparent lors de l'utilisation de la commande Vérification.

12.2.9 EXTERNE

xjpdraw peut exécuter des programmes externes en fournissant comme entrée de ces programme une description des primitives existantes, puis en lisant le résultat du programme (voir page 54 pour plus de détails).

12.2.10 FOLLOW_MOUSE

En mode "Sélection" une primitive peut être translatée avec l'aide de la souris. Cette commande gére ce type de possibilité selon trois possibilités. La commande:

```
# FOLLOW_MOUSE never
```

fera que l'on ne pourra jamais déplacer une primitive avec la souris.

La commande:

```
# FOLLOW_MOUSE always
```

fera que l'on ne pourra toujours déplacer une primitive avec la souris.

La commande:

```
# FOLLOW_MOUSE variable
```

fera que la possibilité de déplacer une primitive avec la souris dépendra d'une action sur la touche L6. Ce mode est le mode par défaut, configuré initialement pour empêcher les déplacements. Une action sur la touche L6 permet de passer au mode déplacement, une action sur L6 permettant de revenir au mode sans déplacement (l'icône "Sélection" indique dans quel cas on se trouve).

12.2.11 FONT

Pour afficher les textes xjpdraw utilise diverses fontes, selon la taille et le style du texte. Cette commande permet d'indiquer le nom de ces fontes. Le programme xfontsel permet de rechercher dans votre installation les fontes disponibles. Le mot-clé FONT est suivi par un autre mot-clé indiquant le nom de la font concernée et sa taille: STANDARD, GRAS ITAL, suivi du fichier de font, comme par exemple:

```
# FONT GRAS10 -adobe-times-bold-r-normal--8-80-75-75-p-47-iso8859-1
```

qui indique que les textes en gras, 10 pt utilise une certaine fontes adobe. Par défaut les fontes utilisées sont:

STANDARD8	-adobe-times-medium-r-normal–8-80-75-75-p-44-iso8859-1
STANDARD10	-adobe-times-medium-r-normal–12-120-75-75-p-64-iso8859-1
STANDARD12	-adobe-times-medium-r-normal–12-120-75-75-p-64-iso8859-1
STANDARD14	-adobe-times-medium-r-normal–14-100-100-100-p-74-iso8859-1
STANDARD16	-adobe-times-medium-r-normal–17-120-100-100-p-84-iso8859-1
STANDARD18	-adobe-times-medium-r-normal–18-180-75-75-p-94-iso8859-1
STANDARD24	-adobe-times-medium-r-normal–24-240-75-75-p-124-iso8859-1
GRAS8	-adobe-times-bold-r-normal–8-80-75-75-p-47-iso8859-1
GRAS10	-adobe-times-bold-r-normal–12-120-75-75-p-67-iso8859-1
GRAS12	-adobe-times-bold-r-normal–12-120-75-75-p-67-iso8859-1
GRAS14	-adobe-times-bold-r-normal–14-100-100-100-p-76-iso8859-1
GRAS16	-adobe-times-bold-r-normal–17-120-100-100-p-88-iso8859-1
GRAS18	-adobe-times-bold-r-normal–18-180-75-75-p-99-iso8859-1
GRAS24	-adobe-times-bold-r-normal–24-240-75-75-p-132-iso8859-1
ITAL8	-adobe-times-medium-i-normal–8-80-75-75-p-42-iso8859-1
ITAL10	-adobe-times-medium-i-normal–12-120-75-75-p-63-iso8859-1
ITAL12	-adobe-times-medium-i-normal–12-120-75-75-p-63-iso8859-1
ITAL14	-adobe-times-medium-i-normal–14-100-100-100-p-73-iso8859-1
ITAL16	-adobe-times-medium-i-normal–17-120-100-100-p-84-iso8859-1
ITAL18	-adobe-times-medium-i-normal–18-180-75-75-p-94-iso8859-1
ITAL24	-adobe-times-medium-i-normal–24-240-75-75-p-125-iso8859-1

Le mot-clé `JPDRAWFONT` permet d'indiquer la fonte utilisée pour les boutons et menus de `xjpdraw`. Par défaut on a:

```
#FONT JPDRAWFONT
-misc-fixed-bold-r-normal--13-100-100-100-c-70-iso8859-1
```

Il est nécessaire d'ajuster cette fonte pour obtenir une disposition convenable des éléments de la fenêtre `xjpdraw`.

12.2.12 FONTMATH

Certains symboles mathématiques dans les textes sont convertis pour apparaître sous leur forme réelle à l'écran. Pour cela on utilise deux fonts mathématiques qui sont par défaut:

```
# FONTMATH -adobe-symbol-medium-r-normal--14-140-75-75-p-85-adobe-fontspecific
# FONTMATH -adobe-times-medium-r-normal--12-120-75-75-p-64-iso8859-1
```

Avec cette commande vous pouvez indiquer d'autres fontes.

12.2.13 FULL_MANUAL

Cette commande permet l'accès à un fichier contenant le fichier Postscript du manuel complet de `xjpdraw`. On indique par exemple:

```
# FULL_MANUAL /u/merlet/manuel.ps
```

La visualisation du manuel nécessite d'avoir mis à jour la commande `GHOSTVIEW` dans le configurateur.

12.2.14 GHOSTVIEW

Cette commande doit indiquer un exécutable qui permet la visualisation de fichiers Postscript. Par exemple si l'on utilise `ghostview` la commande sera:

```
# GHOSTVIEW /usr/bin/gnu/ghostview
```

12.2.15 GHOST

Cette commande permet de compiler un fichier LaTeX, de le convertir en Postscript et de visualiser le Postscript correspondant. Cette commande est nécessaire lorsque l'on utilise la commande `Compiler et visualiser` du menu `FICHIER` de `xjpdraw`. Sa syntaxe est illustrée sur l'exemple suivant:

```
# GHOST dvips -f * >bidon1.ps;/usr/bin/gnu/ghostview bidon1.ps
```

Cette syntaxe correspond à un script Unix. Ici on génére le fichier Postscript `bidon.ps` à l'aide du programme `dvips` puis on visualise ce fichier Postscript à l'aide du programme `ghostview`. Il est impératif que le nom du fichier crée soit bidon1.ps. N'indiquez pas d'ordre de compilation LaTeX dans cette commande, un processus dans ce but tourne tâche de fond et s'en charge.

12.2.16 GHOST_UPDATE

Cette commande permet de mettre à jour le fichier Postscript décrivant le dessin. C'est l'ordre qui est utilisé par la commande `Recompiler pour le previewer` du menu `FICHIER`. Une forme possible est:

```
# GHOST_UPDATE dvips -f * >bidon1.ps&
```

Il est impératif que le nom du fichier crée soit bidon1.ps. N'indiquez pas d'ordre de compilation LaTeX dans cette commande, un processus dans ce but tourne en tâche de fond et s'en charge.

12.2.17 GS

`xjpdraw` permet la visualisation de fichier Postscript externe. Le principe de cette manipulation consiste à utiliser le programme `gs` de Gnu pour créer une image du fichier à inclure, image qui sera affichée dans le canevas de `xjpdraw`. Cette commande permet d'indiquer le nom et le chemin d'accès du programme `gs` comme indiqué dans l'exemple:

```
# GS /usr/bin/gnu/gs
```

12.2.18 LANGUAGE

Cette commande permet d'indiquer si l'on utilise la version française ou anglaise de `xjpdraw`. La syntaxe est:

```
# LANGUAGE french
```

ou

```
#LANGUAGE english
```

La valeur par défaut est `french`.

12.2.19 LATEX_COMMAND

Cette commande a pour but d'indiquer le nom du programme LaTeX, qui sera utilisé pour la compilation des textes. Un exemple est:

```
# LATEX_COMMAND latex
```

Elle est nécessaire si l'on a mis la commande COMPILE_LATEX à 1.

12.2.20 LATEX_VERSION

Cette commande permet d'indiquer quel version de LaTeX on va utiliser: les versions possibles sont `latex2.09` (version par défaut) ou `latex2e`.

```
LATEX_VERSION latex2.09
LATEX_VERSION latex2e
```

Cette commande est utilisée pour définir l'en-tête par défaut des documents LaTeX complets.

12.2.21 LIB

La commande `LIB` indique le répertoire où l'on trouve les différents fichiers optionnels pouvant être utilisés par `xjpdraw`.

```
# LIB /u/merlet/Jpdraw
```

12.2.22 LIB_EXTERNE

Cette commande permet de spécifier le répertoire où se trouve les programmes externes que `xjpdraw` est capable de faire exécuter, voir page 54 pour plus de détails.

12.2.23 MANUAL

Cette commande permet d'indiquer le nom et le chemin d'accès de la documentation en ligne. Dans la distribution standarde le nom de ce fichier est `doc-en-ligne`. On indiquera donc:

```
# MANUAL /u/merlet/Jpdraw/doc-en-ligne
```

12.2.24 PRINTING

Cette commande est utilisée pour compiler et imprimer un fichier LaTeX. Elle est utilisée pour la commande `Compiler et imprimer` du menu `FICHIER`. Sur cette ligne de commande on indique donc un script Unix permettant de réaliser cette fonction comme par exemple:

```
# PRINTING latex %;dvips *
```

Le symbole % sera remplacé à l'exécution par le nom `bidon.ltex` et le symbole * par le nom `bidon` (ici l'impression est assurée par le programme `dvips`). Une commande équivalente serait:

```
# PRINTING latex %;dvips -o bidon.ps *; lpr -Plw bidon.ps
```

si `lw` est le nom de l'imprimante.

12.2.25 PS

Malheureusement il n'y a pas de manière standarde d'inclure un fichier Postscript dans un source LaTeX car cette manipulation dépend du programme qui permet de convertir le fichier `.dvi` en fichier Postscript. Ainsi si l'on utilise `dvi2ps` la syntaxe est:

```
\special{ps:plotfile [fichier Postscript]}
```

Pour permettre une utilisation avec un programme différent on dispose de différentes possiblités définies dans le configurateur. La commande `PS` permet d'indiquer la syntaxe à utiliser. Par exemple pour retrouver la syntaxe du `dvi2ps` on mettrait:

```
# PS \special{ps:plotfile %s}
```

A l'exécution le symbole %s sera remplacé par le nom du fichier à inclure. Dans cette instruction on peut aussi indiquer deux flottants qui seront alors affecté aux dimensions (en cm) du dessin. Ainsi pour le programme `dvitops` on va mettre:

```
# PS \special{dvitops: import %s %fcm %fcm}
```

alors que pour le programme `dvips` on peut utiliser:

```
# PS \epsffile{%s}
```

si l'on utilise les macros `epsf` ou

```
# PS \special{psfile="`zcat %s" hscale=\the\jpdrawx \sp vscale=\the\jpdrawx}
```

si on n'utilise pas les macros `epsf`. Dans ce dernier cas on utilise pleinement les possibilités de dvips. Tout d'abord on suppose que les fichiers Postscript générés par `xjpdraw`sont compressés. L'utilisation de la commande `zcat` permet de décompresser à la volée ces fichiers. Noter que la présence du mot `zcat` dans la commande `PS` fait que `xjpdraw` va automatiquement compresser les fichiers Postscript qu'il crée. Ensuite on utilise la variable `\jpdrawx`. Cette variable est initialisée dans les fichiers LaTeX crées par `xjpdraw`. Par défaut cette variable vaut 100. La partie de la commande avec les mot-clés `hscale, vscale` permet d'ajuster la taille du Postscript inclus, à partir de la valeur de `jpdrawx`. Cette valeur est justement modifiée par la valeur de `unitlength`. Par exemple si dans le corps du document LaTeX on inclut le dessin `toto.ltex` par:

```
\unitlength 0.8 cm
\input{toto.ltex}
```

la variable `\jpdrawx` va valoir 80 et le dessin sera imprimé à 80% de sa taille nominale. Rappelons aussi que si `unitlength` a une valeur négative le dessin sera imprimé de façon à ce que sa dimension maximale (hauteur ou largeur) vaudra la valeur absolue de `unitlength`.

Notons que la syntaxe utilisée doit être compatible avec le type de programme qui est utilisé pour la conversion des fichiers `.dvi` tel qu'indiqué par la commande `DVI->PS`.

Lors de l'écriture d'un document important il peut être intéressant de créer un répertoire ne contenant que les figures. On peut alors placer un configurateur dans ce répertoire de façon à ce que le chemin indiqué pour trouver le fichier Postscript soit automatiquement mis à jour. Par exemple si dans le répertoire `Text` vous avez crée un répertoire `Figures` vous placez dans ce répertoire un configurateur contenant l'instruction:

```
# PS \special{ps:plotfile Figures/%s}
```

Après avoir utilisé la commande `export JPDRAW=configurateur` vous pouvez lancer `xjpdraw`. Les fichiers LaTeX sauvés contiendront bien le chemin d'accès au fichier Postscript: vous pouvez ainsi mettre dans le corps de votre document des références à des dessins du répertoire `Figures` sans vous préoccupez de la localisation des fichiers Postscript.

Une autre manière de faire serait d'indiquer automatiquement le chemin d'accès complet au fichier, en partant de la racine. Pour cela il suffit de remplacer le symbole %s dans la commande par le symbole %S.

12.2.26 PSEXTERNE

Le but de cette commande est le même que pour la commande `PS` mais elle sera utilisée pour inclure les fichiers Postscripts non pas crées par `xjpdraw` mais externes. La syntaxe est identique.

12.2.27 ROTATE_ORDER

Cette command permet de définir l'ordre employé dans les fichiers LaTeX pour la rotation des textes. L'ordre est décrit avec une phrase contenant les mot-clés %r, %R, qui seront remplacés par plus ou moins l'angle de rotation en radian, %d, %D, qui seront remplacés par plus ou moins l'angle de rotation en degré et %s qui sera remplacé par le texte. Par exemple si l'on veut utiliser le package `rotating` de LaTeX2e on pourra mettre:

```
\begin{rotate}{%r} %s \end{rotate}
```

Si cette commande n'est pas définie et que vous avez des textes tournés on utilisera les macros de `lplainext.tex`, qu'il faudra donc inclure dans le corps de votre document.

12.2.28 SELECTION_TYPE

Cet ordre doit être suivi de `icon` ou `creation_and_icon`. Il permet d'indiquer que la sélection d'une primitive ne peut se faire qu'après sélection de l'icône de sélection (c'est le mode par défaut) ou bien que la primitive sélectionnée est toujours la dernière crée.

12.2.29 SLIDE_HEIGHT, SLIDE_WIDTH

Ces ordres sont suivis d'un nombre qui indique la hauteur et la largeur de la page en centimètre, lorsque l'on est en mode transparent

12.2.30 START_SLIDE

La chaîne qui suit cet ordre sera incorporé avant chaque page du document dans le fichier `bidon.ltex` en mode transparent lors de l'utilisation de la commande `Vérification`.

12.2.31 UNDO_DIR

Le mécanisme d'annulation des ordres se fait par écriture à chaque ordre effectué d'un fichier `.macdraw` temporaire et par son rappel si nécessaire. Cette commande à pour but d'indiquer dans quel répertoire sont sauvés ces fichiers. Par défaut le répertoire est `/tmp`. On pourra ainsi mettre :

```
# UNDO_DIR /tmp
```

Le nom des fichiers d'annulation est du type:

```
undo[nombre]_[nom de l'utilisateur].macdraw
```

12.2.32 UNDO_NUMBER

Le mécanisme d'annulation des ordres est automatiquement supprimé lorsque le nombre de primitive est trop importante. Cette commande fixe le seuil. On pourra mettre :

```
# UNDO_NUMBER 200
```

12.2.33 Touches

L'affectation des touches de contrôle `R,L, F` se fait en donnant le nom de la touche suivie de son code numérique (KeyCode) et d'une chaîne de caractères entre quote selon la syntaxe:

```
#[touche][nombre] [code] "[nom]"
```

Ainsi:

```
# R6 51 "Insert"
```

indique que la touche `R6` correspond à la touche du clavier de code numérique (KeyCode) 51 dont le nom est `Insert` (pour obtenir les KeyCode des touches vous pouvez utiliser le programme `xev` ou la commande `xmodmap -pk`). Noter que les aides en ligne prennent en compte le nom réel de la touche.

12.2.34 Vérification

Vous pouvez vérifier votre configurateur en utilisant la commande :

```
xjpdraw -check
```

`xjpdraw` vérifiera ce que vous avez défini dans votre configurateur et s'il toruve des erreurs tentera de vous proposer une solution de substitution. Le diagnostic est imprimé sur la sortie standard. Un traitement particulier est réservé à l'assignation des touches: les KeyCode proposés seront imprimés avec leur `KeySymb` correspondant. Vous pourrez alors vérifier dans le fichier:

```
/usr/include/X11/keysymdef.h
```

que vous avez correctement assigné vos touches.

12.2.35 Exemple

Un exemple de fichier configurateur est le suivant:

```
/*-------------------------------------------------
to get an english version just change french in english
in the following line
---------------------------------------------------------*/
# LANGUAGE french
# LIB /u/merlet/Jpdraw/Configurateur
# BIB /u/merlet/Util_Jpdraw
# DVI->PS dvips
# PS \special{psfile="`zcat %S" hscale=\the\jpdrawx \sp vscale=\the\jpdrawx}
/*-----------------
affectation des touches
------------------*/
# R1 31 "<-"
# R2 35 "->"
# R4 27 "^"
# R5 34 "\/"
# R6 51 "Insert"
# R7 101 "Ins"
# R13 119 "End"
# L1 8 "Stop"
# L2 10 "Again"
# L3 32 "Props"
# L4 33 "Undo"
# L5 56 "Front"
# L6 58 "Copy"
# L7 79 "Open"
# L8 80 "Paste"
# L9 102 "Find"
# L10 104 "Cut"
/*---------------------------------
 clignotement des primitives. Par defaut:
texte (0): 6 ligne(1):20 carre(2):11  ellipse(3):4  ovale(4):4
polygone(5): 11 ou 2 pour les grands
------------------------------------*/
# BLINK 0 4
# BLINK 1 25
# BLINK 2 6
# BLINK 3 3
# BLINK 4 20
# BLINK 5 20 2

/* pour imprimer un fichier */

# PRINTING latex %;dvips *

/* pour visualiser du postscript */

# GS /usr/bin/gnu/gs
# GHOST  dvips -f * >bidon.ps;/usr/bin/gnu/ghostview bidon.ps
# GHOSTVIEW /usr/bin/gnu/ghostview
# GHOST_UPDATE dvips -f * >bidon.ps&
# PSEXTERNE \special{psfile="%S" hscale=\the\jpdrawx \sp vscale=\the\jpdrawx}

/* documentation en ligne */

# FULL_MANUAL /u/Jpdraw/manual.ps
# MANUAL /u/merlet/Jpdraw/doc-en-ligne
```

```
/* compilation des textes */

# COMPILE_LATEX 1
# LATEX_COMMAND latex

/* pour Annuler */

# UNDO_NUMBER 200
# UNDO_DIR /tmp

/* symboles grecs */

# FONTMATH -adobe-symbol-medium-r-normal--14-140-75-75-p-85-adobe-fontspecific
# FONTMATH -adobe-times-medium-r-normal--12-120-75-75-p-64-iso8859-1

/* le repertoire des modules externes */

# LIB_EXTERNE /u/lib/Jpdraw

/* taille des transparent

# SLIDE_HEIGHT 23
# SLIDE_WIDTH 15
```

12.3 Le configurateur annexe

L'utilisation de la commande `BIB` permet d'avoir une librairie de modules communs à un groupe. Mais il peut être aussi intéressant d'avoir la possibilité de définir des modules propres à un utilisateur donné. Ceci peut se faire en définissant une variable d'environnement `LOCAL_JPDRAW` qui indique le chemin d'accès à un configurateur annexe. Ce configurateur ne doit contenir que des ordres de type `BIB`. Les modules apparaissant dans les répertoires correspondant seront alors ajoutés aux modules du configurateur principal. Le nom des répertoires ajoutés est imprimé au lancement du programme.

Chapitre 13

Conclusion

Vous allez sans doute rencontrer quelques problèmes au cours de l'utilisation de `xjpdraw`. Mais soyez indulgent car ce programme devait initialement servir simplement à tracer des droites et des rectangles. Et puis rappelez vous de ces quelques maximes:

-Year: A period of three hundred and sixty-five disappointments.

-43rd Law of Computing:

Anything that can go wr

`xjpdraw` Segmentation violation – Core dumped

-Scott's second Law:

When an error has been detected and corrected it will be found to have been wrong in the first place.

Corollary

After the correction has been found in error it will be impossible to fit the original quantity back into the equation.

Bonne chance!

Chapitre 14

Annexes

14.1 Annexe 1: Le format .macdraw

Pour les initiés ! Les primitives sont stockées dans un tableau de structure du type primit définie par:

```
struct primit {
    int type;
    int visible;
    float para[11];
    char texte[256];
    int num_poly;
    int nbpoint;
    int poly[30][2];
    };
```

où type indique la nature de la primitive.

valeur	nature de la primitive
0	texte
1	ligne
2	rectangle
3	ellipse
4	ovale
5	polygone

dans primit.para, on a successivement:

type	para[]	nature
0	(0,1)	x,y début texte
	(2)	style (0=standard,1=gras,2=italique)
	(3)	taille caractère (0=10pt,1=12pt,2=14pt,3=16pt,4=18,5=24,6=8)
		si image postscript hauteur en pixel
	(4)	1 = texte souligné
		si image postscript numéro de l'image
	(5)	1 si c'est un commentaire
		si image postscript x du début réel de l'image
	(6)	angle de rotation en radian
		si image postscript x de fin réel de l'image
	(7)	1: image postscript, > 1: texte centré dans primit (7)-2
	(8)	si image postscript y de fin réel de l'image sinon largeur réelle du texte
	(9)	nombre de mot dans le texte
		si image postscript hauteur image ppm
	(10)	si image postscript y de fin réel de l'image sinon hauteur réelle du texte
1	(0,1)	x,y début ligne
	(2,3)	x,y fin ligne
	(4,5)	pente en x pente en y (valeur LaTeX)
	(6)	0 ou 1 si pente compatible LaTeX
	(7)	1=vecteur, 2=pointillé 3=double vecteur
		> 3 pointillé =(7)+1, 200 vecteur, 202 et suivant pointillé -3=double vecteur inversé, 400 et suivant pointillé
		300 double vecteur, 302 et suivant, pointillé
	(10)	si ligne de texture numéro de la primitive englobante+1
2	(0,1)	x,y premier coin
	(2,3)	x,y deuxième coin
	(4)	type de trait 1=full 2= carré+texte 3 = carré pointillé+texte
		4= carré pointillé
	(5)	position du texte,0=pas de texte,1=centre,2= a gauche
		3=a droite,4= en haut,5=en bas,6=en haut a gauche,
		7=en haut a droite 8=en bas a gauche,9=en bas a droite
	(6)	la texture grise 1=noir , 100=blanc
	(7)	angle de rotation
	(10)	unité= taille du texte dizaine=style, si >100 souligné
3	(0,1)	x,y centre
	(2,3)	x,y du rectangle englobant (ellipse)
		ou d'un point sur le cercle
	(4)	0=ellipse, 1= cercle
	(5)	texture grise (1=noir, 100= blanc), si < 0 pas de bordure
	(6)	angle de rotation pour ellipse
		si arc , millier=angle départ, million=angle fin
	(7)	1 si cercle avec 2 lignes
		2=fleche t1 vers t2, 3 inverse, 4 double flêche 5 double flêche inversé
	(10)	0 =trait plein, 1=trait pointillé

type	para[]	nature
4	(0,1)	x,y centre
	(2,3)	x,y du rectangle englobant
	(4)	>1= dashed, pointillé = (4)+1
	(5)	1 pour rester compatible avec xjpdraw_v1 si teta=0
	(6)	angle de rotation
	(7)	type:
		0 ovale complet,1 1/2 ovale supérieure
		2 1/2 ovale inférieure,3 1/4 ovale sup,gauche
		4 1/4 ovale sup,droit,5 1/4 ovale inf,gauche
		6 1/4 ovale inf,droit
		si < -2 la même chose que le positif mais avec flêche
5	(0)	1= côté en pointillé
	(1)	0 sinon 2 si c'est une flêche blanche 3= flêche noire
		4 si c'est une parabole
	(2)	len si flêche, 1(2) si parabole a axe vertical(horizontal)
	(3)	dep si flêche, si parabole coeff. a de la parabole
		en unité page
	(4)	wid si flêche, si parabole angle de rotation
	(5)	1= ligne non fermée
	(6)	si flêche pente x (LaTeX) ,si le polygone appartient
		a un tracé crayon alors numéro du tracé
		si parabole x origine de la parabole (unité écran)
	(7)	si flêche pente y (LaTeX)
		si parabole y origine de la parabole (unité écran)
		si polygone général couleur de texture (1=noir,100 blanc)
	(10)	si parabole x(y) fin parabole, si axe horizontal(vertical)
		(unité écran)

Pour la couleur des rectangles:

- gris/couleur avec bordure: para[6]=PEN >0
- noir avec bordure: para[6]=1 PEN=2
- gris/couleur sans bordure: para[6]<0 PEN >0 = en abs
- bordure colorée: para[6]=0 PEN>0
- bordure grise: para[6]=0 PEN>0

Pour la couleur des cercles, ellipses:

- gris/couleur avec bordure: para[5]=PEN >0
- noir avec bordure: para[5]=1 PEN=2
- gris/couleur sans bordure: para[5]<0 PEN >0 = en abs
- bordure colorée: para[5]=0 PEN>0
- bordure grise: para[5]=0 PEN>0

Pour la couleur des polygones généraux:

- gris/couleur avec bordure: para[7]=PEN >0
- noir avec bordure: para[7]=2 PEN=2
- gris/couleur sans bordure: para[7]<0 PEN >0 = en abs
- bordure colorée: para[7]=0 PEN>0
- bordure grise: para[7]=0 PEN>0

Pour toute primitive:
visible=1 si visible 0 si non visible

para[8] donne le style de la ligne 0=fin, 1=épais standard. Si compris entre 0 et 24 l'épaisseur vaut (L+6)/10 mm sinon (31-L)/10 mm

para[9] est égale au nombre de mot dans le texte, il ne doit jamais être nul (exception pour les lignes et les cercles).

dans la structure primit le texte est toujours rempli, si le texte est inutile on place le mot `__vide__`.

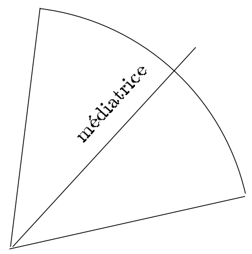
Ces structures sont sauvées dans le fichier `.macdraw` sous la forme:

slide nombre de primitives
(pour chaque primitive)
type visibilité
para[0..11]
(si polygone:)
numero polygone
nombre de point du polygone
coordonnées des sommets du polygone
(fin du cas polygone)
texte
.....
liste des numéros de primitive dans l'ordre du tracé
....
dimensions du dessin
des lignes sans importance
`---`
`couleur`
`---`
une liste d'entiers qui est le code couleur des primitives

Le mot `slide` en tête du fichier n'est présent que pour ceux effectués en mode transparent.

14.2 Quelques exemples de dessin

14.2.1 Texte tourné

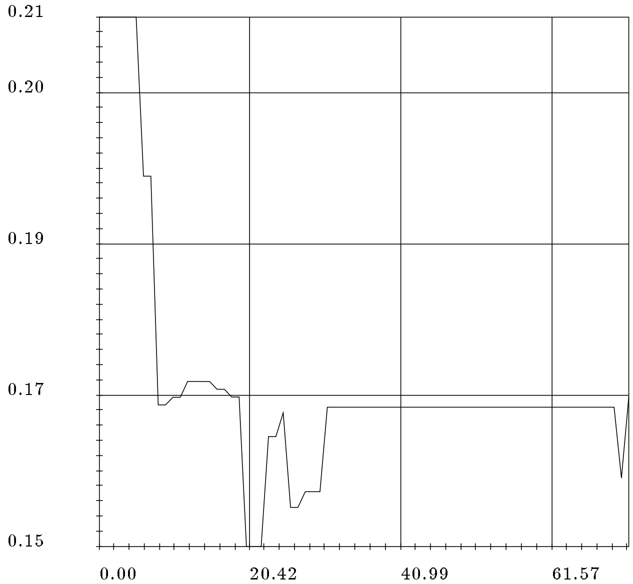


14.2.2 Tracé de fonction

On dispose de 635 couples de coordonnées de points résultant d'une expérience en robotique dans le fichier y.7, mis au format fonction utilisateur (les coordonnées sont précédées de l'instruction `mode polygone -1 10`. On peut alors créer les segments joignant ces points. La commande a été lancée par:

xjpdraw -X 9 -Y 5 -F y.7

qui a crée automatiquement le cadre et le tracé de fonction. Le quadrillage ainsi que les valeurs des abscisses et des ordonnées ont été placés automatiquement ultérieurement.

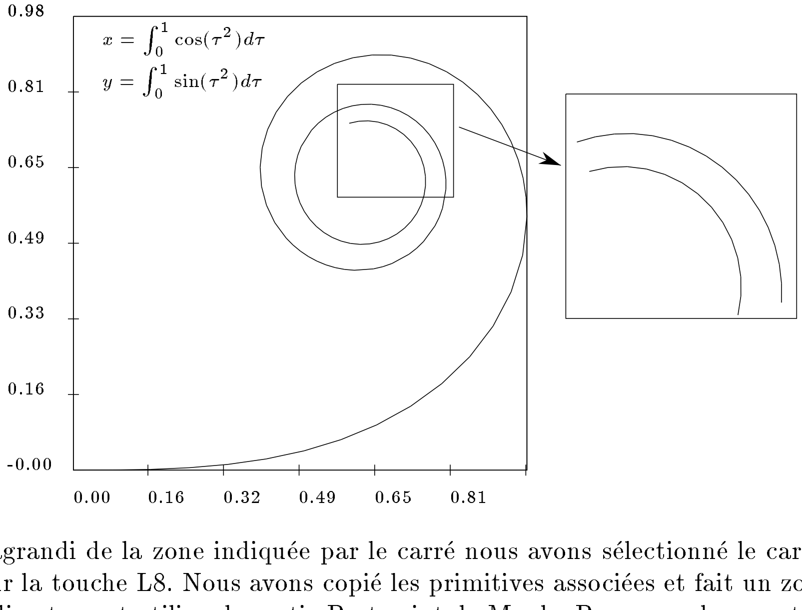


14.2.3 Maple

On désire tracer une clotoïde définie par les équations

$$x = \int_0^t \cos(\tau^2) \qquad y == \int_0^t \sin(\tau^2)$$

Le mode **Formule** de **xjpdraw** ne permettant pas de traiter ce genre d'équations nous avons utilisé Maple pour la dessiner et nous avons récupéré les points du tracé Maple pour en faire une fonction utilisateur.

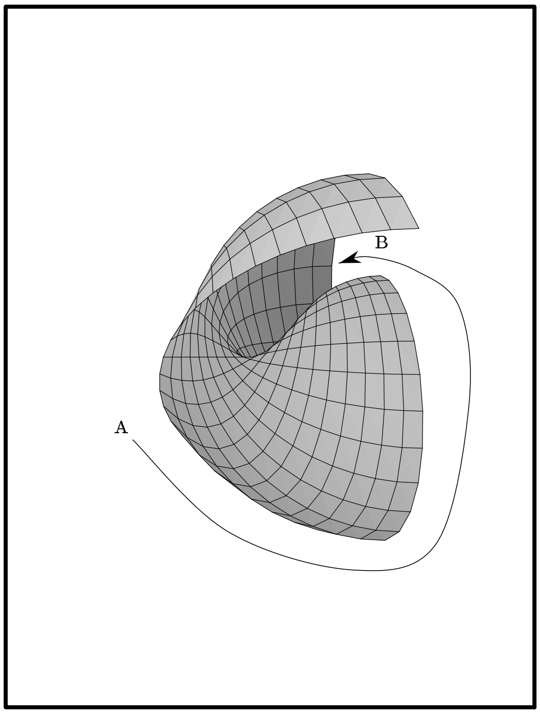


Puis pour faire un agrandi de la zone indiquée par le carré nous avons sélectionné le carré, associé les primitives internes en appuyant sur la touche L8. Nous avons copié les primitives associées et fait un zoom sur les associées.

On aurait pu aussi directement utiliser la sortie Postscript de Maple. Par exemple pour tracer la surface :

$$\begin{aligned} X &= x\sin(x)\cos(y) \\ Y &= x\cos(x)\cos(y) \\ Z &= x\sin(y) \end{aligned}$$

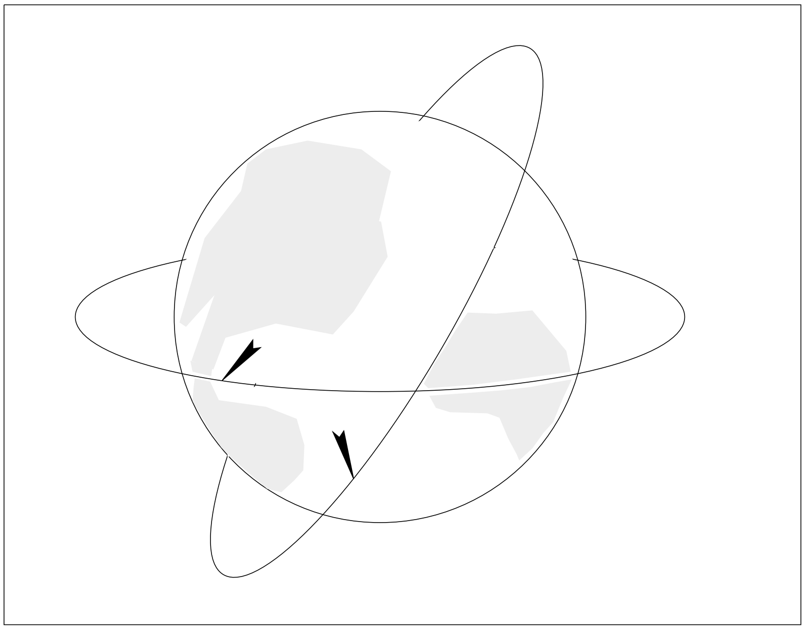
avec $x \in [0, 2\pi]$,$y \in [0, 2\pi]$.



Ici on a directement utilisé le fichier Postscript produit par Maple et on a simplement ajouté la courbe spline allant de A à B.

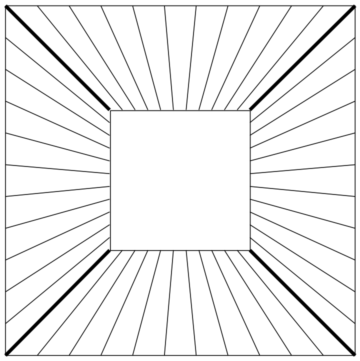
14.2.4 Rotations, satellite et masque

Après avoir défini la Terre, une orbite pour le satellite et son champ de vision on a associé ces 3 éléments, puis copié et soumis à des rotations successives autour du centre de la Terre. La partie cachée des orbites a été supprimée en placant un polygone dessus, que l'on a texturé en blanc sans bordure. On aurait aussi pu utiliser les ciseaux.



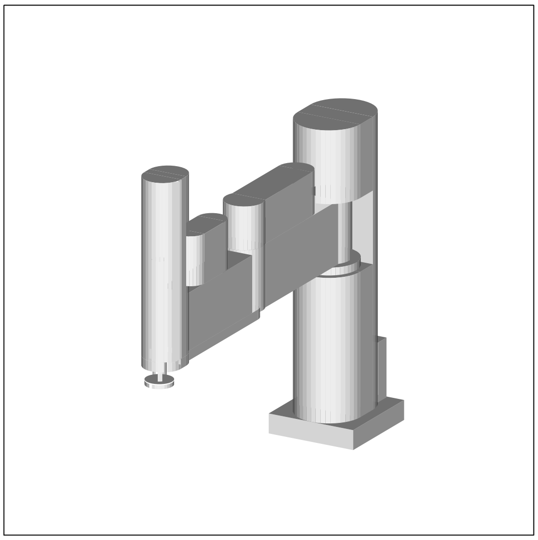
14.2.5 Duplication

Pour créer cette pièce on a tout d'abord généré le carré extérieur en mode rectangle avec l'option carré. On a ensuite copié cette primitive avec un facteur d'échelle (menu `EDITION`). On a ensuite crée les 4 lignes épaisses en mode ligne en activant l'aimant en mode "Sommet". Puis on a associé successivement deux paires de lignes consécutives et l'on a demandé une duplication progressive avec un nombre de 10.



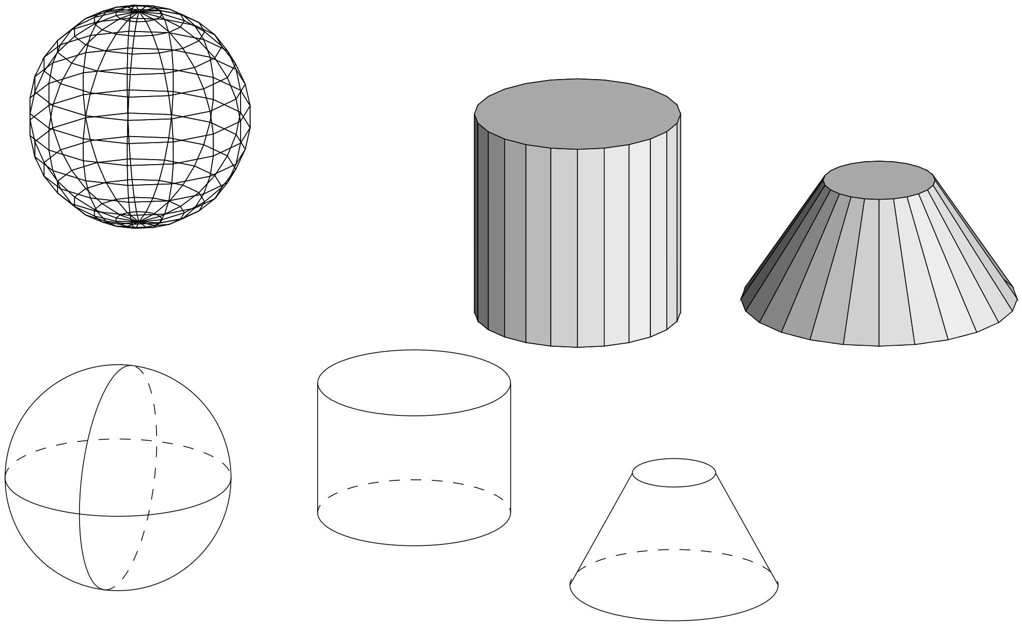
14.2.6 Interface avec un programme de CAO

Le dessin du robot ci-dessous est le résultat de la lecture d'une fonction utilisateur crée par un système de CAO.



14.2.7 Module 3D

Voici quelques dessins que l'on peut obtenir directement en utilisant le module 3D.



14.3 Principales différences avec la version 4.1

- en mode changement de taille d'une ligne possibilité d'utiliser la touche L8 pour avoir les pentes de LaTeX standard
- compilation des textes par une tâche de fond
- possibilité de rajouter des textes en les copiant d'une autre fenêtre
- possibilité d'active ou de deactiver la compilation immédiate des textes
- ajout du bouton "Ajuster la marge" dans aligner
- le mode marque est activé par F10
- en mode marque '-' détruit la dernière marque, '=' les détruit toutes
- pour les translations précises ou l'action de L9 on vérifie que la translation ne met pas de primitives hors cadre
- dans la fenêtre de "Charger" ou de l'album deux boutons de plus: un pour ne pas montrer les fichiers de sauvegarde, l'autre pour les détruire
- dans la fenêtre des fichiers en mode "Charger" ou "Album" tapez un espace ou un tab assure la complétion du nom
- l'ajout de flêche au polygone marche aussi pour les lignes
- CTRL-x est un raccourci pour "Recompiler pour le previewer"
- en mode rotation on peut donner l'angle en littéral
- pour les cercles et les marques cliquer avec le bouton de gauche fait avancer alors qu'avec le bouton du milieu on recule.
- CTRL-i est un raccourci pour la réglure
- en programmation les expressions `function_max_x,function_min_x` donnent les valeurs des maximum et minimum selon l'axe x de la fonction utilisateur chargée. Idem pour l'axe y.
- ajout du mode transparent permettant de créer des documents multi-page.

14.4 Erreurs

14.4.1 Messages au lancement

`# should be followed by instruction`
Dans le configurateur on a trouvé un # non suivi d'une commande
`Attention:la variable d'environnement JPDRAW indique un fichier..`
Votre variable d'environnement JPDRAW pointe sur un mauvais fichier configurateur
`Bad argument to BIB in local configuration file`
Le configurateur local contient un # sans argument
`Bad local configuration file`
Le configurateur local contient une commande inconnue
`Bad number after UNDO_NUMBER in configuration file`
UNDO_NUMBER doit être suivi d'un nombre
`Erreur:ces fonts ne sont pas a definir`
Vous avez défini des fonts de nom incorrect dans le configurateur
`Erreur:chaine trop longue dans header_end.ltex:on l'ignore`
Le fichier header_end.ltex est incorrect
`Erreur:dans header_end.ltex (probablement il manque un CR)`
Le fichier header_end.ltex est incorrect
`Erreur:dans header_end.ltex:ignore (il manque un % dans le special)`
Le fichier header_end.ltex est incorrect
`Erreur:dans le fichier de configuration, ligne`
Vérifier la ligne
`Erreur:hdcp-> chaine trop longue (>255)-> pas de copie d'ecran:`
Erreur dans le script hdcp
`Erreur:mauvais fichier de font`
Dans le configurateur on a définit une fonte incorrecte
`Erreur:Pas de script hdcp pour copie d'ecran`
L'opération `copie écran` n'est pas possible car pas de script hdcp pour copie d'ecran dans la directory `LIB` du configurateur. Mettez en un
`Erreur:Pas de script hdcpc pour copie ecran couleur`
L'opération `copie écran` n'est pas possible car pas de script hdcp pour copie d'ecran dans la directory `LIB` du configurateur. Mettez en un
`Error: argument to BEZIER <=0`
L'argument de BEZIER doit être positif
`Error in BLINK argument`
Le premier argument de BLINK doit être compris entre 0 et 5
`Error in BLINK`
L'instruction BLINK doit être suivi de deux ou de trois nombres
`Error in BLINK...must be followed by an int`
BLINK doit être suivi d'un entier
`Error in BLINK 5:must be followed by two int`
BLINK 5 doit être suivi de deux entiers
`Error in configuration file, bad argument to BIB`
BIB doit être suivi d'une chaîne
`Error in configuration file, F key`
F doit être suivi d'un nombre et d'une chaîne, il manque le nombre
`Error in configuration file, L key`
L doit être suivi d'un nombre et d'une chaîne, il manque le nombre
`Error in configuration file, missing F key name`
F doit être suivi d'un nombre et d'une chaîne, il manque la chaîne
`Error in configuration file, missing L key name`
L doit être suivi d'un nombre et d'une chaîne, il manque la chaîne
`Error in configuration file, missing R key name`
R doit être suivi d'un nombre et d'une chaîne, il manque la chaîne
`Error in configuration file, R key`

R doit être suivi d'un nombre et d'une chaîne, il manque le nombre

`Error in CONTEXT`

La commande CONTEXT du configurateur doit être suivie d'un nom de fichier

`Error in CONTEXT: bad file`

Le nom suivant la commande CONTEXT n'est pas le nom d'un fichier existant

`Error in` $DVI->PS$

$DVI->PS$ doit être suivi par une chaîne

`Error in FOLLOW_MOUSE: invalid chain`

FOLLOW_MOUSE doit être suivi de always, never, variable

`Error in FOLLOW_MOUSE`

FOLLOW_MOUSE doit être suivi d'une chaîne

`Error in FULL_MANUAL`

FULL_MANUAL doit être suivi du chemin d'accès au manuel

`Error in GHOSTVIEW`

GHOSTVIEW doit être suivi du nom de la commande ghostview

`Error in LANGUAGE`

LANGUAGE doit être suivi de la chaîne "french" ou "english"

`Error in UNDO_DIR`

UNDO_DIR doit être suivi du nom du répertoire où est effectué la sauvegarde des fichiers d'Annuler

`Error in UNDO_NUMBER`

UNDO_NUMBER doit être suivi du nombre maximum de primitives neutralisant l'ordre ANNULER

`Error: no argument to BEZIER`

BEZIER doit être suivi par une chaîne

`Error: unknown argument to` $DVI->PS$

L'argument de $DVI->PS$ n'a pas été reconnu

`Error:Bad argument to PS`

L'argument de PS est incorrect

`Error:LIB not followed by argument`

LIB doit être suivi par le nom du répertoire des modules

`Error:no argument to PS`

PS doit être suivi par une chaîne

`LANGUAGE should be french or english in configuration file`

`Local libraries should contain only # BIB`

`Negative number after UNDO_NUMBER in configuration file`

UNDO_NUMBER doit être suivi d'un nombre positif

`Too many local libraries, ignoring`

Le configurateur local contient trop de libraries

`Unknown command in configuration file`

Une commande inconnu a été trouvée

`Warning: no name associated to key`

Une chaîne de caractère doit être associée au nom d'une touche.Vérifier le configurateur

`You have to indicate a configuration file for running xjpdraw..`

Avant de lancer `xjpdraw` il faut que la variable d'environnement JPDRAW indique le configurateur

`Error in EXTERNE (name)`

voir syntaxe de EXTERNE

`Error in EXTERNE (input)`

voir syntaxe de EXTERNE

`Error in EXTERNE (type input)`

voir syntaxe de EXTERNE

`Error in EXTERNE (output)`

voir syntaxe de EXTERNE

`Error in EXTERNE (type output)`

voir syntaxe de EXTERNE

`MINITEL should be followed by 4 int`

La commande minitel doit être suivi de 4 entiers indiquant la taille de la fenêtre

`Error in SELECTION_TYPE`
cette commande doit être suivie de `icon` ou `creation_and_icon`
`Error in SLIDE_HEIGHT`
cette commande doit être suivi d'un nombre indiquant la hauteur d'un transparent
`Error in SLIDE_WIDTH`
cette commande doit être suivi d'un nombre indiquant la largeur d'un transparent

14.4.2 Messages généraux

`Annulation non disponible (repertoire inconnu)`
Les fichiersd'annulation n'ont pu être crées, modifier le UNDO_DIR de votre configurateur
`Annulation non disponible (trop d'objet)`
Il y a trop de primitive pour annuler l'opération
`Annulation non disponible (vous l'avez choisi)`
Vous avez suppriméla possibilité d'annulation, voir le menu PAGE
`Attention: erreur probable dans votre texte`
Votre texte contient une erreur latex
`Attention:cannot open font`
Problème à l'ouverture d'une fonte, l'affichage seule sera incorrecte
`Attention:je ne peux ouvrir des fonts`
Certaines fonts du configurateur n'existe pas
`Attention:Le nombre de primitive est tres proche du nombre maximum`
`xjpdraw` approche du nombre maximum de primitive, essayer de vider le presse-papier
`Attention:Un point d'un polygone est hors de la page`
un polygone sort du cadre, ramener le avec la commande ramener les primitives hors-cadre
`Attention:vous n'avez pas mis d'extension, j'ai rajoute l'extension .ltex`
Lors de la commande sauver vous avez indiqué un fichier sans extension. `xjpdraw` a rajouté l'extension .ltex
`Erreur:Associer seulement le texte et la primitive`
Pour centrer un texte il faut que seul le texte et la primitive où l'on veut le mettre soit associés
`Erreur:attention au nombre maxi de primitives!`
L'opération va créer trop de primitives
`Erreur:Cannot allocate so much primitive, sorry`
Vous avez indiqué un chiffre trop gros dans l'option -M
`Erreur:Cannot find image file`
Lors de l'incorporation d'un fichier Postscript externe il y a eu un problème de création d'image, vérifier si `gs` arrive à lire votre fichier
`Erreur:Cannot incorporate postscript file,GS not defined`
Vous avez demandé d'incorporer un fichier Postscript externe mais la commande GS du configurateur n'est pas défini. Modifier votre configurateur
`Erreur:Ce n'est pas un texte`
L'opération nécessite que la primitive soit un texte
`Erreur:Desole je ne sais pas creer de spline avec ces points de controle`
Le programme de calcul du spline a échoué. Modifier la position des points de contrôle.
`Erreur:help indisponible, verifier MANUAL dans le configurateur`
Vous avez demandé l'aide en ligne mais dans le configurateur l'instruction MANUAL est incorrecte. Vérifier le configurateur
`Erreur:il faudrait qu'il y ait une primitive selectionne!`
Cette opération nécessite de sélectionner une primitive
`Erreur:Il faut associer les polygones a decouper`
Vous avez demandé de découper les polygones en arêtes, mais il faut associer les polygones que vous voulez couper
`Erreur:il faut au moins deux points pour un polygone`
Vous avez tenté de créer un polygon à un seul sommet
`Erreur:Il faut d'abord definir une texture pour la rendre permanente!`
Vous avez essayé de rendre une texture permanente mais vous ne l'avez pas défini. Utiliser le bouton définir et recommencer
`Erreur:il faut selectionner une primitive..`
Cette opération nécessite de sélectionner une primitive
`Erreur:Il n'y a pas de texte a centrer`
Il n'existe pas de primitive de texte dans les associées. La commande centrer un texte le nécessite

`Erreur:Impossible de couper ici`
Le ciseau ne peut pas couper la primitive en ce point
`Erreur:Impossible de couper les textes`
Vous avez essayé d'utiliser le ciseau sur un texte. Utilisez plutôt le bouton L10 suivi de emacs
`Erreur:Impossible de tout couper, trop de primitives seraient crees`
Vous avez demande de couper les polygones associés mais le nombre de primitive qui seraient obtenu après coupure va être trop important
`Erreur:installation incorrecte(pas de header):probleme avec VERIFICATION!`
On n'a pas trouvé de fichier préambule dans le répertoire `LIB`. Les commandes de vérification du menu FICHIER ne fonctionneront pas
`Erreur:Je ne peux pas creer de perpendiculaire ici`
Il n'existe pas de perpendiculaire à la primitive au point sélectionné
`Erreur:Je ne peux pas creer de tangente`
Il n'existe pas de tangente à la primitive au point sélectionné
`Erreur:je ne peux pas ecrire dans cette directory ou trop de fichiers ouverts`
On ne peut pas sauver dans le répertoire courant
`Erreur:Je ne peux pas mettre une fleche sur ce polygone trop petit`
Le polygone est trop petit pour que l'on puisse calculer sa tangente en ce point
`Erreur:Je ne sais centrer que dans un rectangle ou un cercle`
Vous avez demandé de centrer un texte dans un primitive qui n'est pas un rectangle ou un cercle
`Erreur:Je ne sais pas centrer dans un rectangle tourne`
Pour centrer un texte dans un rectangle il ne faut pas que le rectangle soit tourné
`Erreur:La selectionnee n'est pas un polygone`
L'opération demandé impose que la primitive sélectionnée soit un polygone
`Erreur:La selectionnee n'est pas un texte`
L'opération demandé impose que la primitive sélectionnée soit un texte
`Erreur:le fichier n'existe pas`
Le fichier demandé n'existe pas
`Erreur:Le point n'est pas sur la primitive selectionnee`
Le point de coupure pour le ciseau doit être proche de la primitive que l'on veut couper
`Erreur:le polygone a deja le nombre maximum de point`
On ne peut pas créer un nouveau sommet au polygone
`Erreur:Must be a number >1`
Le nombre que l'on doit fournir doit être > 1
`Erreur:Must give a number`
On attend un nombre dans cette opération
`Erreur:No ellipse created`
Problème lors de la création d'une ellipse par 4 points, les 4 points ne sont pas en position générale
`Erreur:No such file`
Le fichier n'a pas été trouvé
`Erreur:nombre primitives dessin album > 1500,essayer de vider le presse-papier`
L'ajout de l'album ferait que le dessin aurait trop de primitives
`Erreur:Not a polygon`
L'opération se fait uniquement sur un polygone
`Erreur:Not possible on an external postscript file`
Opération impossible sur un Postscript externe
`Erreur:Operation impossible sans Extensions`
Opération impossible sans les extensions, passez en mode Extension en cliquant dans le bouton de la fenêtre des menus
`Erreur:Pas assez de point de controle pour la creation d'un Bspline`
Impossible de créer un B-spline avec les points indiqués
`Erreur:Pas assez de point de controle pour la creation d'un spline`
Impossible de créer un spline avec les points indiqués
`Erreur:Pas d'associees`
L'opération nécessite qu'il y ait des primitives associées
`Erreur:pas de changement de forme pour cette primitive!`
On a essayé de changer les dimensions d'un primitives pour laquelle c'est impossible
`Erreur:Pas de fichier de ce nom`

Le fichier demandé n'existe pas

`Erreur:Pas de marque pour cette touche!`

Vous avez appuyé sur un touche chiffre mais le il n'existe pas de marque associé à cette touche

`Erreur:Pas de polygone associe`

L'opération demandée impose qu'il y ait des polygones associés

`Erreur:pas de premier plan pour ce type`

La primitive sélectionnée ne peut pas être mis au premier plan

`Erreur:pas de primitive selectionnee`

Cette opération nécessite de sélectionner une primitive

`Erreur:Pas de primitive selectionnee`

L'opération demandée impose qu'il y ait une primitive sélectionnée

`Erreur:Pas de rotation des postcripts externes`

Il est impossible de faire tourner un postscript externe

`Erreur:pas de texture pour ce type de primitive`

On ne peut pas texturer la primitive sélectionnée

`Erreur:Pas de zoom pour les poscripts externes`

On ne peut pas zoomer un postscript externe. Vous pouvez seulement faire un zoom de toute la figure en changeant le unitlength

`Erreur:Pas d'homothetie sur ce type de primitive`

Copier avec homothétie n'est pas possible sur ce type de primitive

`Erreur:Plus de place pour definir une nouvelle marque!`

Vous avez déjà défini 10 marques, qui est le nombre maximum

`Erreur:Probleme d'ouverture de fichier`

Le fichier demandé n'existe probablement pas ou vous avez trop d'applications ouvertes

`Erreur:Probleme possible pour la compression du .ps`

La commande compress n'a pas fonctionné, compressez votre fichier à la main

`Erreur:seul les fichiers d'extension .macdraw sont lisibles`

On a essayé de lire un fichier d'extension différente de .macdraw

`Erreur:The command GS is not defined, cannot include postscript`

Vous avez demandé d'inclure un fichier postscript externe mais la commande GS n'a pas été défini dans le configurateur. Modifier le configurateur

`Erreur:The number must be >0`

On doit fournir un nombre $>$ à 0

`Erreur:trop de caractere!`

On a essayé d'introduire un texte trop long

`Erreur:Trop de ligne de texture`

La texture va créer trop de lignes

`Erreur:trop de point dans le trace au crayon`

On a fait un tracé à la souris trop longtemps

`Erreur:trop de point d'intersection pour cette ligne de texture!`

la coupure de la texture s'est mal faite

`Erreur:Trop de point pour creer le polygone`

Les associées définissent plus de 30 points, nombre de sommets maximum pour un polygone

`Erreur:trop de point pour trouver le polynome`

On a donné trop de point de contrôle pour un polynôme

`Erreur:Trop de primitive pour couper`

On ne peut pas créer de primitives en coupant car cela créerait trop de primitives

`Erreur:Type incompatible pour la duplication`

L'opération dupliquer impose que les primitives associées soit du même type

`Erreur:Une primitive hors cadre,je ne peux pas centrer.`

Le dessin est trop gros pour que l'on puisse le centrer, utilisez le zoom

`Erreur:Vous devez associer au moins 2 primitives, la 1ere etant la fixe`

Pour aligner il faut associer les primitives à aligner

`Erreur:Vous devez d'abord associer le texte puis la primitive`

Pour centrer un texte dans une primitive associez les deux éléments

`Erreur:Vous devez d'abord selectionner un rectangle ou un polygone`

On peut associer les primitives internes seulement à un rectangle ou à un polygone

`Erreur:Vous devez d'abord selectionnez la primitive a texturer`

Pour texturer une primitive il faut d'abord la sélectionner

`Erreur:Vous devez d'abord selectionnez la primitive a tourner`

Pour faire tourner une primitive il faut d'abord la sélectionner

`Erreur:Vous n'avez pas donne soit la formule,soit X0, soit X1, soit le pas!`

Pour tracer une courbe analytique il faut donner: la gamme sur le x et le pas sur le x

`Erreur:vous ne pouvez pas mettre un angle nul ici`

pas d'angle nul pour une texture inclinée

`Erreur:vous ne pouvez pas sauver dans un fichier d'extension .macdraw`

On a tenté de sauver le dessin dans un fichier d'extension .macdraw. Il faut donner une autre extension

`help indisponible, verifier MANUAL dans le configurateur`

`Impossible pour les postscript externes`

Opération non permise

`Je ne trouve pas le fichier image`

L'incorporation d'un postscript externe passe par la création d'une image ppm, qui na pu être crée

`La commande GS n'est pas defini, je ne peux rien faire`

Pour cette opération il est nécessaire d'utiliser GS, vérifier votre configurateur

`Probleme durant le calcul de la taille du textte`

Le calcul des dimensions du texte s'est mal passé

`Probleme lors du calcul des dimensions`

Le calcul des dimensions du texte s'est mal passé

`Vous devez associer les lignes`

Pour découper des lignes il faut les associer d'abord

`Vous devez donner un nombre`

`Zoom trop grand`

Après un tel zoom la taille du dessin excédera les dimensions de la fenêtre

14.4.3 Messages en mode Fonction utilisateur

`Attention:mode connecte ou deconnecte sans point`

les modes deconnecte ou connecte doivent être suivis de points

`Attention:mode spline sans argument`

pas de point de contrôle pour un spline

`Attention:pas de texte dans un mode texte`

le mode texte doit être suivi d'un texte

`Attention:polygone a moins de deux points`

un mode polygone doit être suivi d'au moins deux points

`Attention:rayons non egaux en mode arc, je prend le premier comme rayon`

le mode arc doit être suivi des coordonnées du centre du cercle puis de 2 valeurs identiques qui sont le rayon du cercle

`Attention:style incorrect en mode texte, je met 10 pt`

le mode texte est suivi d'un nombre qui indique le style. Le nombre trouvé ne correspond pas à un style connu

`Erreur:Trop de point pour un spline : je tronque`

On a donné trop de point de contrôle pour définir un spline

`Erreur:Vous devez d'abord selectionner le rectangle ou vous allez placer le trace`

Avant de tracer une fonction il faut sélectionner le rectangle dans lequel se fera le tracé

`Erreur:coordonnees incorrectes en mode polygone`

des coordonnées sont incorrectes dans le mode

`Erreur:d'argument:definition incorrecte pour une fonction`

les arguments de `xjpdraw` au lancement sont incorrects

`Erreur:dans -X (compris entre 0 et 14.5)`

L'argument de -X doit être compris entre 0 et 14.5

`Erreur:dans -Y (compris entre 0 et 14.5)`

L'argument de -Y doit être compris entre 0 et 14.5

`Erreur:dans la position du texte`

vérifier les coordonnées de ce mode texte

`Erreur:dans le centre en mode cercle`

vérifier les coordonnées qui suivent le mode cercle

`Erreur:dans le rayon en mode cercle`

Vérifier le rayon, deuxième ligne après le mode cercle

`Erreur:dans origine d'une fonction (dessin hors cadre 14.5x14.5)`

les arguments -O, -X, -Y doivent conduire à un dessin inclu dans un rectangle de 14.5 cm au maximum

`Erreur:le format des echelles est incorrect`

Vérifier le format des échelles que vous donnez pour la fonction

`Erreur:le format donne n'est pas correct,je le remplace par 6.2`

Le format des textes est incorrect

`Erreur:mauvais centre en mode arc`

Vérifier les coordonnées du centre qui suivent le mode cercle

`Erreur:mauvais parametre d'angle en mode arc`

Vérifier les angles du mode arc

`Erreur:mauvais rayon en mode arc`

Vérifier le rayon du cercle du mode arc

`Erreur:mauvais type en mode polygone`

Vérifier le nombre qui suit le mode polygone

`Erreur:mauvaise hauteur ou largeur en mode rectangle`

Vérifier les dimensions du mode rectangle

`Erreur:mode inconnu`

Un mode du fichier fonction est inconnu

`Erreur:pas de parametre en mode rectangle`

Vérifier les paramètres du mode rectangle : on doit avoir origine et largeur et hauteur

`Erreur:trop de points dans un polygone (mettez -1 dans le nombre)`

Le nombre indiqué de points d'un polygone ne correspond pas au nombre de points fournis

`Erreur:trop de points de controle dans un spline`

On a fourni trop de point de contrôle dans un spline

14.4.4 Messages en mode Programmation

`Erreur:Pas de point souris en mode invisible!`

Vous êtes en mode invisible et dans le programme vous avez demandé l'introduction de point à la souris

`Erreur:Trop de points souris`

On a demandé trop de point souris

Index

D

Table des matières