



HAL
open science

Notice d'utilisation JPdraw : version 3.3

Jean-Pierre Merlet

► **To cite this version:**

Jean-Pierre Merlet. Notice d'utilisation JPdraw : version 3.3. RT-0139, INRIA. 1992, pp.109. inria-00070029

HAL Id: inria-00070029

<https://inria.hal.science/inria-00070029>

Submitted on 19 May 2006

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

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

INRIA

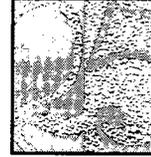
UNITÉ DE RECHERCHE
INRIA-SOPHIA ANTIPOLIS

Institut National
de Recherches
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
BP 105
78153 Le Chesnay Cedex
France
Tél (1) 39 63 55 11

Rapports Techniques

1 9 9 2



ème
anniversaire

N° 139

Programme 4
Robotique, Image et Vision

NOTICE D'UTILISATION JPDRAW, VERSION 3.3

Jean-Pierre MERLET

Mars 1992



★ RT . 8 1 3 9 ★

Notice d'utilisation JPdraw , version 3.3

JPdraw User's manual, version 3.3

Jean-Pierre MERLET

Résumé

Nous décrivons l'utilisation d'un éditeur de dessin fonctionnant sur SUN dans l'environnement suntools ou sur n'importe quelle station de travail dans l'environnement X. Cet éditeur génère 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) et fournit de nombreux outils pour leurs manipulations.

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

Abstract

We describe the use of a graphic editor for the SUN workstation in the suntools environment or any workstations in the X-window environment. This editor generates \LaTeX files for the text and Postscript files for the drawing which can be included in a \LaTeX file.

It enables to create various geometric primitives (lines, texts, boxes, polygons, ellipses, ovals and miscellaneous curves) and various tools to manipulate them.

Furthermore two meta-languages for the description of graphics in an external file are understandable by this software, enabling the creation of drawings as results of various programs (for instance CAD system or image processing).

Chapitre 1

Le premier dessin

1.1 Introduction

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:

`merlet@cygnusx1.inria.fr`

Any suggestions or comments are to be sent at the above E-mail adress

JPdraw est un outil de dessin orienté géométrie fonctionnant dans l'environnement SUNTOOLS et, de manière expérimentale, sous X-Window. Il est même possible d'utiliser dans certains cas JPdraw avec un simple terminal non graphique. Il permet de tracer des figures simples sur un écran de SUN et génère des fichiers pour l'éditeur de texte \LaTeX (en utilisant l'environnement Picture, page 101 du Manuel \LaTeX).

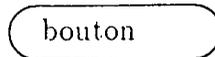
Un dessin JPdraw est constitué de plusieurs éléments appelés "primitives". Ce sont des lignes, des textes, des carrés etc...Le but de cet éditeur est de manipuler ces primitives à l'aide de la souris et de quelques touches fonction. Le nombre maximum de primitives est de 1500. La description suivante de cet outil correspond à la version SUNTOOLS. Quelques différences existent avec la version X-Window et feront l'objet d'un chapitre.

1.2 Pour commencer: lignes, rectangles, ellipses

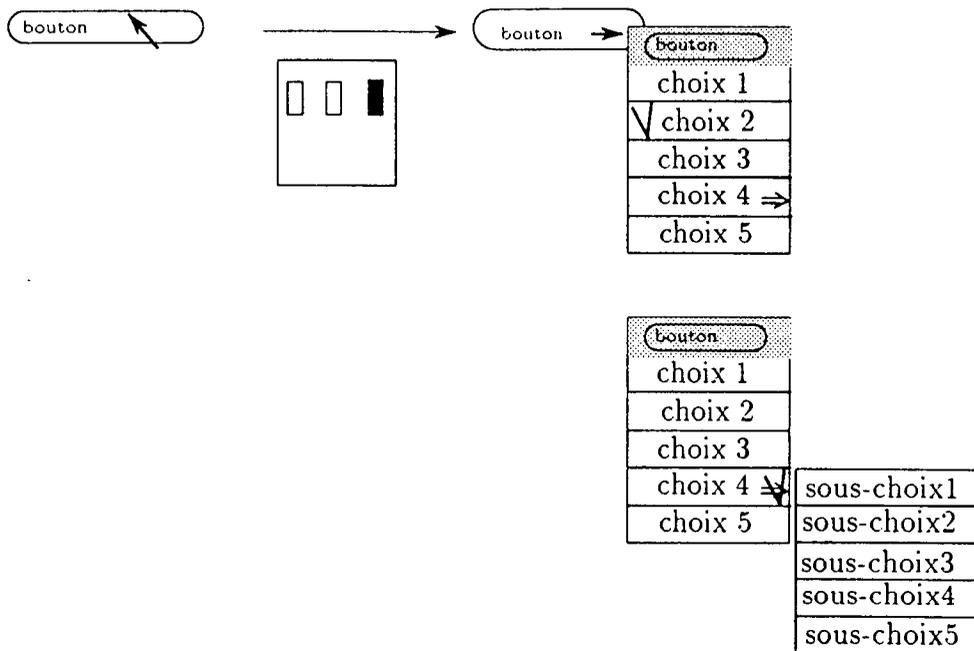
1.3 Boutons, Menus et Icônes

1.3.1 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. 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 :



Le menu d'un bouton apparaît avec le bouton de droite

Figure 1.1: Usage des boutons

Ce menu se compose d'une case noircie rappelant le nom du bouton puis de différentes cases disposées selon un axe vertical indiquant les choix possibles pour ce bouton (figure 1.1).

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.

Bouton sans menu: exécution avec le bouton de gauche

Si un menu complet apparait on peut sélectionner le choix désiré en plaçant la flèche 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 1.2). Parfois un sous-choix est possible: la case comporte alors une flèche horizontale tournée vers la droite et le sous-menu apparait en plaçant la souris dans la prolongation de la case .

Sous-menu

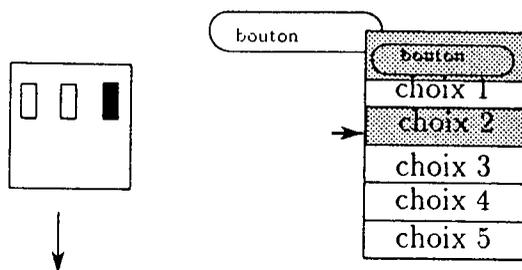


Figure 1.2: Choix dans un menu

Pour valider la sélection il suffit de relacher le bouton de droite.

Les icônes

1.3.2 Icônes

Fondamentalement les icônes sont identiques aux boutons dans leur manipulation. 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. Les icônes peuvent avoir elles aussi des menus. Nous verrons que les icônes dans JPdraw passent en négatif lorsqu'elles sont sélectionnées.

1.3.3 Remarques

- après avoir fait apparaître le menu d'un bouton ou d'une icône le déplacement de la souris ne sert qu'à 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.
- Si l'on clique successivement avec le bouton de gauche dans un bouton avec menu on va exécuter les différentes commandes présentes dans l'ordre où elles apparaissent dans le menu sans que le choix apparaisse à l'écran. Une certaine prudence s'impose donc.

*Fenêtre de
texte*

1.3.4 Les fenêtres de JPdraw

*Fenêtres
actives*

Dans JPdraw 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 du fichier courant). Ces fenêtres ne sont actives que si la souris est placée à l'intérieur de la fenêtre. Lorsque la fenêtre est active un petit curseur triangulaire noir clignote. Si elle n'est pas active ce curseur est fixe et gris (figure 1.3).

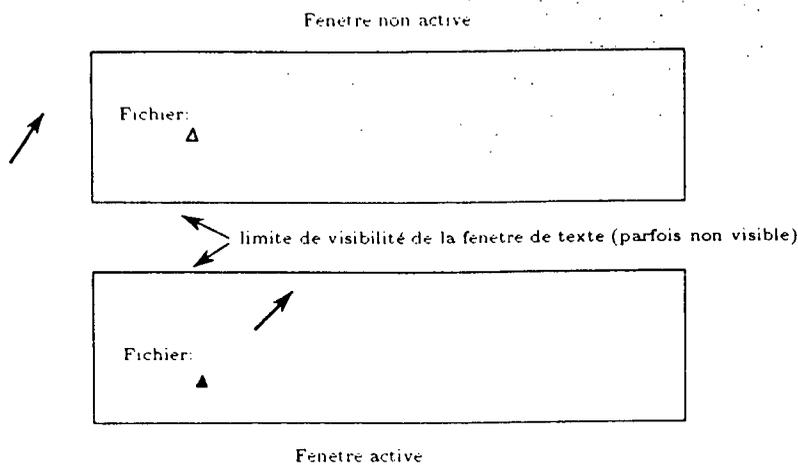


Figure 1.3: fenêtre de texte active ou non

Ces fenêtres sont en général utilisées pour communiquer un texte au programme. Pour valider le texte envoyé il faut taper la chaîne de caractères suivie d'un retour chariot. Pour annuler la commande il suffit de taper seulement un retour chariot.

Ayant défini les éléments qui vont permettre de contrôler les actions de JPdraw on va maintenant lancer le programme.

1.4 Lancer le programme

Le programme est invoqué par la commande `jpdraw`. La commande `jpdraw` est paramétrable, comme nous le verrons dans la section page 81, mais dans un premier temps cela n'a pas d'importance.

*Présentation
de JPdraw*

1.4.1 Tableau de bord

Après avoir lancé la commande JPdraw apparaît un graphique composé de 5 éléments (Figure 1.4):

- un tableau de menus (en haut, horizontal)
- un tableau de d'icônes (à gauche, vertical)
- un canevas (où seront tracées les figures)
- un mini-buffer où seront imprimés les messages émis par JPdraw (en bas horizontal)
- un mini-buffer contenant un éditeur de texte de type emacs (en bas horizontal)

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

*Raccourci
pour le choix
des icônes*

dans le canevas. Le raccourci est indiqué dans l'icône. Par exemple pour tracer des segments de droite on utilise la touche F3. Attention les icônes qui n'ont pas de numéro sont des icônes avec menu.

Le canevas comporte des "scrollbars" qui permettent de visualiser les différentes parties du dessin. Le principe d'utilisation des scrollbars est le suivant:

-en se plaçant dans le coin supérieur gauche, bande horizontale on fait glisser le dessin vers la gauche en cliquant le bouton de droite (c'est logique!); et réciproquement avec le bouton de gauche. En se plaçant dans la bande verticale on fait monter le dessin en cliquant à gauche et descendre en cliquant à droite. En cliquant au milieu on fait buter le dessin.

Le mini-buffer va afficher les informations sur les commandes, les erreurs de manipulation, le nombre de primitive du dessin et éventuellement la position de la souris. Celui contenant l'éditeur **emacs** va permettre d'éditer les primitives de texte.

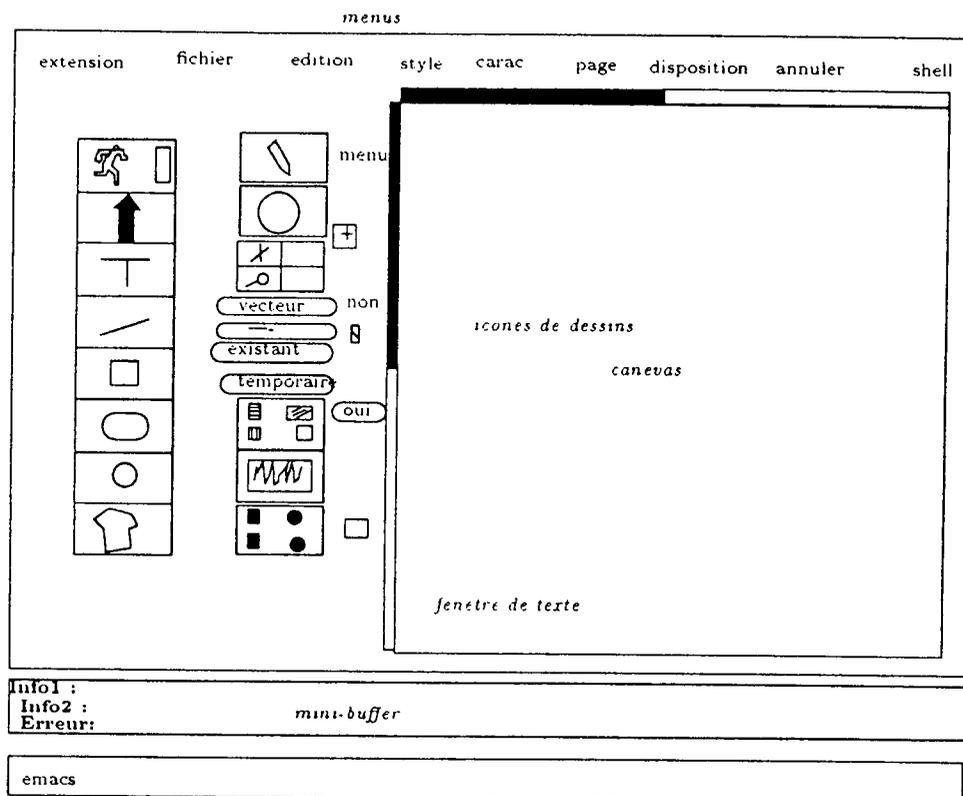


Figure 1.4: Tableau de bord

1.5 Le tracé de primitives

Le choix du type de primitives que l'on va tracer s'effectue à l'aide des icônes. Nous allons donc examiner successivement les icônes qui permettent la création de lignes, de rectangles et d'ellipses, sans se préoccuper des options possibles. 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**.

1.6 Ligne et style de trait



Permet de tracer des lignes et des vecteurs.

Principe: sélectionner l'icône, si l'on veut des vecteurs cliquer sur le bouton vecteur (le **NON** à côté du bouton passe à **OUI**, si on continue à cliquer il passe à "2 X" et dans ce cas les vecteurs auront une flèche aux deux extrémités).

Cliquer à l'origine du segment avec le bouton de gauche et maintenir ce bouton appuyé. En déplaçant la souris le segment apparaît. Lorsque le segment voulu est obtenu relâcher la souris.

Le bouton pointillé permet de choisir le style des traits: fins, épais, pointillés fin ou pointillés épais ainsi qu'éventuellement l'épaisseur des traits (c'est un bouton avec menu). Le choix fin ou épais est valable pour toutes les lignes du dessin. La petite icône apparaissant à côté de ce bouton indique le style courant. Nous verrons en détail page 22 comment manipuler le style des traits.

Les lignes

*Création des primitives:
avec le bouton de gauche*

Le style des lignes

1.7 Rectangle



Ce mode permet de tracer des rectangles.

Sélectionner l'icône. Se déplacer dans le canevas, le curseur apparaît. 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 relâcher une fois la bonne position déterminée.

Les rectangles

1.8 Ellipse



Les ellipses

Permet de générer des ellipses et des cercles. Dans un premier temps on se bornera au tracé des ellipses. Le centre de l'ellipse est déterminé par le point où l'on appuie sur le bouton de gauche de la souris pour la première fois. Maintenir ce bouton appuyé et déplacer la souris. L'ellipse est contenue dans le rectangle défini par le centre et la position actuelle de la souris.

1.9 Comment sauver dans un fichier

A ce stade vous devez savoir maintenant créer des lignes, des ellipses, et des rectangles. Nous allons maintenant sauver le dessin créé et l'imprimer. Pour cela nous allons utiliser une instruction d'un menu, le MENU FICHIER et plus précisément un sous-choix de ce menu.

Préalablement il est nécessaire de savoir que JPdraw peut générer le dessin sous 3 formes différentes:

- en utilisant les possibilités graphiques très limitées de \LaTeX . Deux fichiers sont créés.
- en utilisant une bibliothèque d'ordre \LaTeX qui font appel à des procédures Postscript. Là aussi deux fichiers sont créés. Le dessin obtenu est approximatif par rapport à ce qu'il y a sur l'écran.
- 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. Trois fichiers (un \LaTeX , un Postscript) sont créés. L'inconvénient est qu'il n'existe pas de manière standard d'inclure du Postscript dans un source \LaTeX .

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

1.9.1 Les fichiers générés

*Fichiers créés
par JPdraw*

JPdraw génère deux types de fichier avec les extensions `.ltx` (usuellement) et `.macdraw` (par défaut `xx.ltx` et `xx.macdraw`). Les fichiers avec l'extension `.ltx` 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 .

*Fichiers
.macdraw*

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

1.9.2 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**. En maintenant le bouton de la souris appuyé placer

celle-ci dans la case **Sauver** et déplacer vous sur la droite. On voit alors apparaître un sous-choix qui contient l'instruction **Sauver**. Placer votre souris dans ce sous-choix; celui-ci devient alors noire. Relâcher le bouton de la souris.

Vous allez alors voir apparaître dans le mini-buffer successivement deux curseurs qui vont indiquer le pourcentage de primitives sauvées dans des fichiers. Une fois que ces curseurs ont disparu, sortez de **JDraw**. Vous allez alors constater la présence dans la directory courante de deux fichiers de nom **xx.ltex** et **xx.macdraw**. Ces fichiers contiennent la description **L^AT_EX** et **JDraw** de votre dessin. Pour en être convaincu tapez la commande **jpdraw xx.macdraw**. Vous allez voir apparaître à nouveau votre dessin.

1.9.3 Sauver dans un fichier donné.

*Sauver dans
un fichier
donné*

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

Pour cette opération on utilise la deuxième sous-choix de l'instruction **Sauver** du menu **FICHIER** qui est **Sauver Sous**. Effectuer maintenant la même manipulation que précédemment en s'arrêtant dans la case **Sauver Sous**.

Une fenêtre texte apparaît dans le canevas, en haut. Mettre la souris dans cette fenêtre (le curseur triangulaire doit devenir noir) et introduire le nom complet du fichier, qui doit contenir une extension différente de **.macdraw** (ici **toto.ltex**), puis taper **Return**. **JDraw** va alors générer les deux fichiers **toto.ltex** et **toto.macdraw**. Vous allez à nouveau voir apparaître les curseurs et vous allez retrouver dans le fichier **toto.ltex** la description **L^AT_EX** de votre dessin.

Le nom du fichier courant (extension **.ltex**) apparaît dans la bande noire du cadre **JDraw**.

Remarque

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

1.9.4 Ecrire dans un fichier existant

*Si le fichier
existe déjà...*

A ce stade vous avez donc un fichier **toto.ltex** et un fichier **toto.macdraw**. Que se passe t'il si vous tentez d'écrire un nouveau dessin sous ce nom? Tentez l'expérience en lançant la commande **jpdraw 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 l'instruction ou l'annuler.

Remarque:

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

D'une manière générale pour les opérations impliquant un nombre élevé de primitives et qui ne nécessite pas une opération à la souris (sauver dans un fichier, translater un grand nombre de primitives associées, redessiner le dessin) il est prudent de placer celle-ci à l'extérieur du canevas. Cette opération permet d'accélérer sensiblement les opérations.

1.10 Sauvegarde automatique

*Sauvegarde
automatique*

Une sauvegarde automatique est effectuée régulièrement. Si le nom du fichier courant est **toto** on sauve le fichier sous format **.macdraw** sous le nom **#Ctoto.macdraw**. Un message indiquant qu'une sauvegarde automatique est en cours apparaît dans le mini-buffer. Le temps entre les sauvegardes automatiques est ajusté en fonction du nombre de primitives. Toutefois on peut le fixer à l'aide d'un menu (voir page 43).

1.11 Charger un fichier

*Charger un
fichier .mac-
draw*

1.11.0.1 A l'intérieur de \mathcal{P} draw

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

Une fenêtre texte apparaît dans le canevas, introduire le nom complet du fichier et taper **Return**. Si la fenêtre clignote et que la sonnerie retentit c'est que le fichier demandé n'existe pas. On peut omettre l'extension **.macdraw** mais la chaîne de caractères ne doit pas contenir d'autres extensions.

*Charger au
lancement du
programme*

1.11.0.2 Au lancement du programme

Au moment du lancement du programme on peut donner comme argument à \mathcal{P} draw un nom de fichier **.macdraw**. Là aussi l'extension **.macdraw** n'est nécessaire que si le nom contient plusieurs extensions.

L'album

1.11.0.3 Album

La commande **Album** du menu **FICHER** permet de rajouter au dessin courant le dessin contenu dans un fichier **.macdraw** déjà existant. Après avoir sélectionné l'album une fenêtre s'ouvre où l'on indique le nom du fichier à récupérer avec l'extension **.macdraw**. On peut omettre l'extension **.macdraw** si la chaîne de caractères ne contient pas d'autre extension. Le dessin s'affiche alors. Les primitives de l'album sont associées (voir page 25) pour que l'on puisse les déplacer en bloc facilement. Il est donc prudent de dissocier les primitives existantes préalablement.

1.12 Imprimer

*Les
dessins stan-
dard de $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$*

1.12.1 Les possibilités graphiques de $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$

A ce stade des opérations vous avez maintenant un fichier $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ qui contient la description de votre dessin. Il ne reste plus qu'à l'imprimer. Là les choses se compliquent légèrement. En effet nous avons vu que le fichier $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ crée contient simplement un environnement $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ de type **picture** (le fichier crée par \mathcal{P} draw n'est donc pas compilable directement mais doit être inclus dans un source déjà existant). $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ permet dans cet environnement certains ordres graphiques mais très limités par rapport aux possibilités des imprimantes.

En effet on ne peut y 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 ± 6 (± 4 pour les vecteurs) et premier entre eux)

Pour augmenter les possibilités graphiques on a défini un fichier particulier: *les extensions*.

1.12.2 Les extensions

Nous avons vu que l'on pouvait définir des ellipses avec $\text{\textbackslash JPdraw}$ alors que $\text{\textbackslash LATEX}$ ne connaît pas en standard ce type de primitive. Ceci s'explique simplement: il existe, fourni avec $\text{\textbackslash JPdraw}$, un fichier de nom *lplainext_dvi2ps.tex* (*lplainext.tex* en abrégé), qui est un fichier *d'extensions*. Dans ce fichier on a défini des nouvelles commandes $\text{\textbackslash LATEX}$ qui remplacent et complètent les ordres graphiques de $\text{\textbackslash LATEX}$.

Plus exactement les commandes définies dans le fichier *lplainext.tex* font appel à la possibilité qu'offre $\text{\textbackslash LATEX}$ d'insérer du Postscript dans un fichier. Les commandes définies dans *lplainext.tex* font simplement appel à des primitives Postscript définies dans le fichier *lplainext.psm*.

L'utilisation de ces extensions présentent cependant deux inconvénients. Un des inconvénients est que les sources générées ne sont alors plus compatibles avec les logiciels de pré-visualisation comme *xdvi* (mais le sont pour d'autres previewers). Le deuxième est que la sortie doit se faire sur une imprimante fonctionnant sous Postscript.

*Les nouvelles
commandes
de dessin*

1.12.3 Installation et utilisation des extensions

Pour pouvoir utiliser ces extensions il faut fournir à $\text{\textbackslash LATEX}$ le fichier *lplainext.tex* et à l'intérieur de celui-ci indiquez où se trouve le fichier *lplainext.psm*.

Je décrirais ultérieurement comment doit se faire proprement l'installation de ces fichiers mais nous allons le faire de manière simple pour commencer. Copiez les fichiers *lplainext* dans votre directory courante et modifiez dans le fichier *lplainext.tex* la ligne:

```
\def\PostScriptCommandFile{lplainext.psm}
```

en mettant le bon chemin d'accès (celui de votre directory courante). Dans votre source $\text{\textbackslash LATEX}$ (celui qui va contenir votre dessin) mettez un `\input {lplainext}` après le `\begin{document}`.

1.12.4 Incorporez votre dessin

1.12.4.1 Insérer un dessin par un éditeur de texte

Le fichier $\text{\textbackslash LATEX}$ tel qu'il est créé par $\text{\textbackslash JPdraw}$ commence par un `\begin{picture}` et se termine par un `\end{picture}`. Il est insérable tel quel dans le fichier source avec votre éditeur de texte.

*Insérer
le dessin dans
un source
 $\text{\textbackslash LATEX}$*

1.12.4.2 Insérer un dessin par `\input`

Une autre possibilité est d'utiliser l'instruction \LaTeX `\input` comme par exemple `\input{toto.tex}` qui prend le fichier `toto.tex` et le met dans le source. Si vous ne mettez pas d'extension dans l'argument de `\input`, \LaTeX prend automatiquement l'extension `.tex`.

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

1.12.5 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 plaçant l'instruction `"\unitlength x cm"` préalablement au dessin. Ceci est fait automatiquement dans les fichiers créés par JPdraw. Changer l'unité de longueur permet de faire un agrandi ou une réduction de votre dessin (sauf pour les textes) avec le même source.

1.12.6 La commande Verification Latex

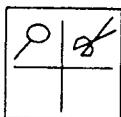
Un fichier, un
dessin:
`bidon.tex`

Il est parfois souhaité, en cours de création d'un dessin, d'imprimer ce dessin seul sur une page. La sous-choix **Verification Latex** de l'instruction **Verifier** du menu **FICHER** permet justement de créer automatiquement un fichier de nom `bidon.tex` qui contient un préambule \LaTeX , le fichier \LaTeX de description du dessin et une fin de document. Ceci est fait simplement par concaténation du fichier `header_latex`, fourni avec JPdraw et qui contient un préambule \LaTeX standard, avec un fichier temporaire dans lequel JPdraw sauve le dessin courant et un fichier `header_end.tex` qui contient une fin de document \LaTeX . Pour que cette commande soit effective il est cependant nécessaire d'installer correctement JPdraw.

1.12.7 Imprimer

Une fois ces opérations terminées il ne reste plus qu'à compiler le fichier `bidon.tex` pour le sortir. Si l'installation des extensions est correcte vous allez voir apparaître à l'écran lors de la compilation des messages du type **Latex extensions enabled** quand il rencontrera un dessin fait à l'aide de JPdraw. Pour l'impression il est nécessaire d'utiliser le programme `dvi2ps` fourni avec JPdraw. L'utilisation d'un autre programme que celui-ci conduit à des résultats aléatoires. Nous verrons qu'il existe cependant des moyens de contourner cette difficulté.

1.13 Utilitaires



Cette icône avec menu permet d'utiliser deux utilitaires. Tout d'abord la loupe qui permet d'obtenir une vue agrandie d'une partie du dessin (voir page 31). Puis les ciseaux qui permettent de couper en un point une primitive (voir page 31). Enfin il y a les menus pour la construction de primitives 3D, dont nous parlerons dans un chapitre ultérieur.

1.13.1 Le module 3D

Dans le menu "Utilitaires" un certain nombre de possibilités de dessin 3D sont offertes. Il s'agit de création de sphère, de parallépipède, de cylindre et de tronc de cône. Attention ces primitives ne sont pas des primitives au sens JDraw mais en sont simplement constituées. On ne peut donc pas les manipuler dans leur globalité après création.

1.13.1.1 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 sliders, qui définissent les angles de vues ou bien de quatre boutons qui permettent de choisir les vues prédéfinies "Perspective", "Dessus", "Face", "Cote".

Ensuite on a à choisir le type de représentation pour les surfaces à l'aide d'un panel de cycle qui permet de choisir soit "Facettes Ombrees", "Facettes" ou "Symbolique".

Dans le premier et le deuxième cas la surface sera représentée par des facettes polygonales. Si le mode est "Facettes Ombrees" les polygones auront une texture en niveau de gris dont l'intensité est fonction de leur position par rapport à l'angle de vue. 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 slider qui apparaît dans ce panel (on y donne le nombre de facettes).

Dans le mode "Facettes Ombrees" 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 représenté par la représentation symbolique classique des primitives, les arêtes cachées apparaissant en pointillé.

1.13.1.2 Sphère

Cliquer avec le bouton de gauche de la souris au centre de la sphère puis en un autre point pour indiquer le rayon.

1.13.1.3 Cube

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).

1.13.1.4 Cylindre

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).

1.13.1.5 Tronc de cône

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).

1.14 Quitter



Sortir

Permet de quitter le programme. Demande confirmation de la sortie.

Chapitre 2

Des primitives plus complexes

On va maintenant introduire quelques primitives plus complexes.

2.1 Texte



Les textes

Permet de placer un texte dans le dessin. On va définir d'abord 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 **JPdraw**). Les lignes de texte commençant par un **%** sont considérées comme des commentaires pour le fichier **L^AT_EX** (ce qui est utile pour indiquer dans le fichier **L^AT_EX** ce que représente le dessin). Ces textes sont interprétés comme du **L^AT_EX** : on peut donc introduire, par exemple, des formules mathématiques complexes.

Placer votre souris à la position voulue (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. Pour terminer déplacez la souris. Les caractères d'effacement sont le backspace et le delete. Après la fin d'entrée du texte vous le verrez apparaître dans le mini-buffer emacs et une boîte en pointillé est tracée autour du texte. Elle indique approximativement la place occupée par le texte après compilation **L^AT_EX**. Pour ce qui concerne l'édition de texte après création reportez vous à la page 29.

Si vous tapez un retour chariot dans votre texte le curseur de la souris se déplace à la ligne suivante.

Les caractères affichés à l'écran doivent correspondre approximativement au style et à la taille choisie si vous disposez des fonts du programme dans votre système. Mais il est possible d'utiliser les fonts disponibles pour obtenir le même effet (voir page 79 pour l'installation de vos fonts personnelles). Si vous n'avez pas les fonts qui sont utilisées par défaut dans le programme et que vous n'avez pas installé vos fonts personnelles le système va utiliser la font

courante. Mais même si le style choisi n'apparaît pas dans le dessin, dans le fichier \LaTeX les fonts seront respectées si vous n'êtes pas en mode math. En effet on rajoute au texte dans le fichier \LaTeX les commandes qui permettent d'obtenir le style choisi. Par exemple le texte "qwerty", affiché à l'écran en gras, sera sauvé sous la forme $\{\text{\bf qwerty}\}$. Bien entendu vous pouvez écrire dans le texte même le style utilisé puisque le texte sera compilé.

2.2 Taille des textes en sortie

Les textes ne sont pas vraiment "whisivig"

Du fait que \LaTeX est un éditeur typographique on ne peut pas garantir que la place prise par le texte dans le dessin correspondra à celle occupée après compilation. Ceci est d'autant plus vrai que le texte est tracé dans le canevas à sa taille réelle alors que les primitives de dessins sont approximativement deux fois plus grandes.

Une bonne manière de vérifier toutefois si le texte occupe la place correcte est de passer le dessin en taille réelle, en utilisant la commande **Taille Reelle** du menu **PAGE**. Les primitives occupent alors à l'écran leur taille réelle alors que les primitives de texte ne subissent pas de changement de taille.

Mais bien sûr cette indication est très rudimentaire (par exemple si vous avez écrit des formules mathématiques) dans la mesure où le texte est compilé. Ceci reste un des gros inconvénients de JPdraw mais aussi un de ces avantages qui est de garder la puissance de \LaTeX pour le traitement de texte.

Par contre JPdraw conserve un aspect "whisivig" pour un certain nombre de caractères spéciaux de \LaTeX . Ce sera le cas pour la majorité des caractères accentués et des symboles mathématiques de \LaTeX . Ceci se passe de la manière suivante: si vous tapez à l'écran le texte `accentu'e` vous verrez s'afficher, après avoir tapé le dernier "e" le texte `accentué`. Pour les symboles mathématiques si vous tapez `Pr'\ecision $ \mu $` vous verrez alors s'afficher `Précision $ \mu $`. En effet JPdraw dispose de deux fonts réservés pour ces symboles spéciaux, les reconnaît dans le texte et affiche le caractère correspondant. Il ne doit pas y avoir cependant plus de 20 symboles spéciaux dans une ligne de texte (au-delà les symboles ne sont plus convertis) et les symboles mathématiques doivent être séparés par des blancs pour être reconnus. Par contre dans la fenêtre **emacs** le texte original reste conservé. La table de l'appendice page 91 donne la liste des symboles traités par cette méthode. Enfin ces caractères spéciaux sont systématiquement écrit en 10 pt pour attirer l'attention sur le fait qu'ils sont traités de manière particulière.

2.3 Polygone



Les polygones

Trace des courbes polygonales fermées ou non.

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 cliquer sur le bouton de droite. 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 points dans un polygone! Si vous dépassez ce nombre la fenêtre clignote et la ligne générée compte le nombre maximum de point.

Pas plus de 30 point dans un polygone!

2.4 Cercle



Permet de générer des ellipses mais aussi des cercles. Après avoir sélectionné cette icône mettez la souris dans le canevas et tapez sur la touche fonction L2. Vous verrez alors apparaître à la place de l'ellipse noir un cercle. Un menu apparaît si l'on clique sur le bouton de droite avec la souris dans l'icône.

Les cercles

Ce menu permet de choisir la manière dont on va définir le cercle.

Les façons de créer un cercle

- 1^{er} choix: cliquer avec le bouton de gauche au centre de l'ellipse ou du cercle, déplacer la souris en maintenant le bouton de gauche appuyé. La figure apparaît, relâcher au bon moment.
- 2^{ème} choix: pour les cercles seulement:
inverse du précédent, on donne d'abord un point sur le cercle puis le centre du cercle.
- 3^{ème} choix: on va donner deux points sur le cercle puis son centre. Cliquer successivement avec le bouton de gauche pour introduire les points sur le cercle, la médiatrice apparaît, placer le centre sur cette médiatrice en cliquant.
- 4^{ème} choix: on donne 3 points sur le cercle en cliquant successivement aux endroits voulus avec le bouton de gauche.
- 5^{ème} choix: permet d'obtenir des cercles avec à l'intérieur deux lignes à 45 degré. C'est l'idéal pour les diagrammes de commande! Attention le cercle est donné dans le mode extérieur puis centre.

Pour obtenir à nouveau les ellipses il suffit d'appuyer sur L4.

2.5 Arc de cercle

On peut créer des arcs de cercle de la manière suivante. Créez tout d'abord un cercle complet en utilisant le 3ème menu de l'icône cercle (tracé avec une médiatrice). Une fois que le cercle est défini vous pouvez supprimer l'arc de cercle entre les deux premiers points cliqués en appuyant sur la touche L6 ou son complémentaire en tapant sur L8. Notez qu'en mode vecteur ou double vecteur les arcs seront munis de flèches.

Les arcs de cercle



2.6 Ovale

Les ovales

Permet de générer des ovales ou des parties d'ovale.

Principe: sélectionner l'icône. Avec le bouton de droite faire apparaître le type de dessin 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 flèche.

- pour l'ovale complet: cliquer pour indiquer la position du centre de l'ovale, maintenir cliquer et se déplacer avec 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 trigo à partir du point de départ. Pour les 1/2 ovales si vous êtes en mode vecteur deux flèches vont apparaître aux extrémités. Elles seront présentes dans le dessin \LaTeX .

Attention les bords arrondis de l'ovale sont très approximatifs (leur réalisation en \LaTeX dépend du système) aussi ne vous servez pas de ces bords comme référence. Lorsque l'on rentre pour la première fois dans ce mode on force le type à ovale complet.

2.7 Marques



Les marques

Permet d'obtenir des primitives prédéfinies dans un dessin (appelées des *marques*), avec la possibilité de définir ses propres marques.

2.7.1 Les marques standards

Les marques standards disponibles sont:

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 dans l'icône puis faire apparaître le menu en cliquant le bouton de droite. Des icônes 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. La petite icône à côté de l'icône marque indique le type de marque courante.

définir ses
propres mar-
ques

2.7.2 Définir et utiliser une marque

Pour un dessin où un certain nombre de primitives vont être utilisées constamment 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**. Cliquer alors sur les primitives que vous voulez définir comme des marques. A chaque marque est alors associée un chiffre entre 0 et 9 (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.

2.8 Le crayon



On utilise ici la souris comme un crayon pour tracer une courbe de forme quelconque. Il existe quatre modes de fonctionnement dans ce cas. Pour faire apparaître les modes il suffit d'appuyer avec le bouton de droite lorsque la flèche de la souris est dans le bouton à côté de l'icône du crayon (bouton marqué **Menu**). Un menu apparaît avec les différentes options. Notez que pour obtenir sur le papier un tracé strictement similaire (et même meilleur) à celui apparaissant sur l'écran il est nécessaire de sauver le dessin en Postscript.

Le crayon

2.8.1 Dessin à la souris (Souris)

Pour tracer une courbe à la main. On produit un ensemble de points qui correspondent aux positions de la souris échantillonnées à environ 1/40 de seconde. Ces points 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.

Tracé à la main

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 clique alors sur le bouton de droite. La courbe passe par les points de contrôle.

Spline

2.8.3 B-splines

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

B-spline

2.8.4 Approximation polynomiale (Polynome)

On procède de la même manière que précédemment pour introduire un maximum de 6 points. Le système trace alors une courbe polynomiale passant le plus près possible des points entrés à la souris.

Courbe polynomiale

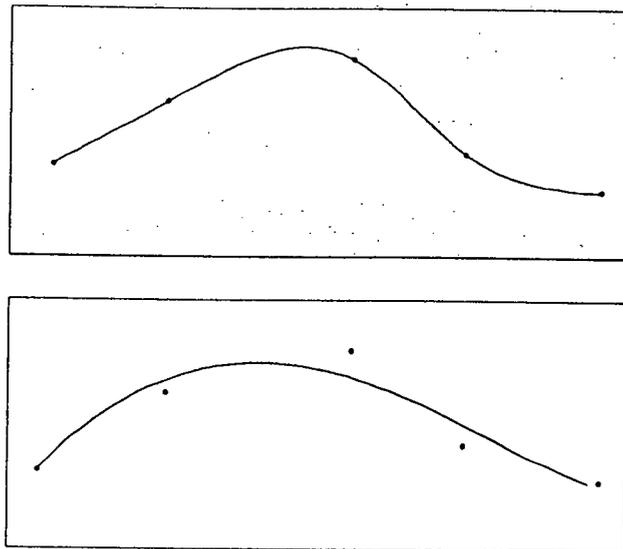


Figure 2.1: Courbes Spline et polynomiale obtenues à partir de points

D'une manière générale il convient d'utiliser le mode **Spline** quant on dispose d'un nombre important de points de passage et le mode **Polynome** pour créer des courbes à partir d'un faible nombre de points.

Parabole

2.8.5 Parabole (Parabole)

Les courbes précédentes sont obtenues par approximation avec des segments de droite. $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ travaillant en entier et ne sachant pas tracer des segments de droite trop courts cela conduit à des courbes qui ne sont pas très lisses (sauf si l'on sauve le dessin en Postscript, voir page 32). On peut remédier à cela en utilisant ce mode. Cette primitive permet de générer des arcs de parabole très lisses en pointillé ou non (figure 2.2).

Pour générer cette primitive on clique avec le bouton de gauche pour indiquer l'origine de la parabole et on maintient ce bouton appuyé en déplaçant 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 relâcher le bouton.

Cette commande utilise un tracé par points qui est très gourmande. Le nombre maximum de points que peut tracer $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ dépend de votre installation. On a supposé que tous les systèmes pourrait effectuer un tracé comportant un maximum de 500 points mais si cette valeur est trop grande on peut la modifier (voir la section 11.1).

Attention:

Ce type de primitive implique pour $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ des calculs assez lourds. La taille mémoire utilisée peut alors parfois dépasser la capacité permise. Pour remédier à cela une première solution consiste à diminuer, dans le fichier $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$, le premier argument des instructions $\backslash\text{bezier}$ qui indique le nombre de point utilisé pour tracer la courbe. Une deux-

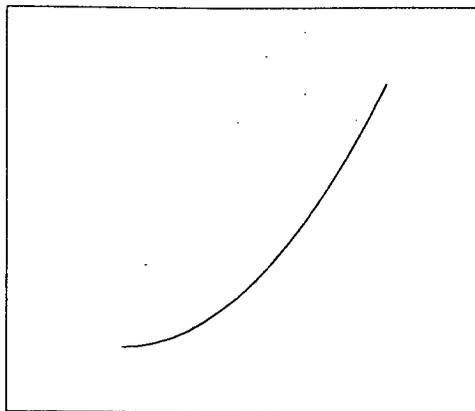


Figure 2.2: Arc de parabole

ième solution consiste à diminuer le nombre de point par centimètre utilisé par JPdraw pour générer ces courbes (voir page 81). Nous verrons par la suite une troisième manière de faire. Bien entendu on ne saurait trop recommander l'utilisation directe du Postscript pour cette primitive.

2.9 Usage avancé des créations de primitives

Nous allons donner ici les possibilités complètes de création de primitives en revenant sur ce que nous avons déjà dit.

2.9.1 Ligne

On peut forcer la pente d'une ligne à être compatible avec celle permise par le \LaTeX standard en maintenant la touche L8 appuyée lors de la création de la primitive. C'est utile en particulier pour tracer des lignes parfaitement horizontales ou verticales. Cela marche aussi pour les vecteurs.

2.9.2 Rectangle

2.9.2.1 Tracé d'un carré

Pour forcer le tracé d'un carré appuyer sur L2 une fois dans le canevas (le mot `carré` apparaît en place de `rectangle` à côté de l'icône). Pour revenir au mode rectangle tapez sur L4.

2.9.3 Polygones particuliers

*Les polygones
particuliers*

2.9.3.1 Tête de vecteur

On crée ici un polygone qui ressemble à une tête de vecteur. Pour cela, en mode polygone tapez sur L3 puis il suffit de cliquer sur les points indiqués dans l'icône pour faire apparaître le polygone en question. Une tête de vecteur est considérée comme un polygone: on peut donc l'annuler par la touche R5 et annuler un point par la touche R4. La touche L10 permet de revenir au polygones généraux.

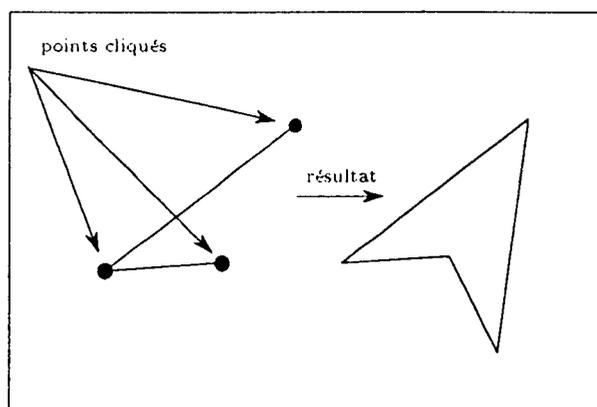


Figure 2.3: Création d'une tête de vecteur

2.9.3.2 Polygone régulier

Polygone régulier

Si vous appuyer sur la touche L9 après avoir sélectionné l'icône polygonale vous pourrez alors tracer des polygones réguliers (un polygone dont tous les côtés sont de longueur égales). 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.... Pour terminer le polygone régulier cliquer avec le bouton de droite. Si vous ne terminez pas le polygone en cliquant sur le bouton de droite ou du milieu le polygone ne sera pas pris en compte. La touche L10 permet de revenir au polygones généraux.

2.10 Le style des traits des primitives

Le style des traits

Le bouton pointillé à côté de l'icône des lignes 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 **Regler l'épaisseur des traits**. Au départ le style épais est fixé à une épaisseur d'environ 0.7mm.
- pointillé: les traits sont fins et en pointillé.

- pointillé épais: les traits sont pointillés, de l'épaisseur fixée par la commande **Regler 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 24 styles de pointillé. En utilisant cette commande on fait apparaître un panel où les différents pointillés sont représentés. Si l'on place la souris aux environs d'un pointillé il clignote. Si on clique ensuite sur le bouton du milieu le style de pointillé est choisie et le panel disparaît. Pour que le pointillé soit utilisé il est nécessaire que le mode pointillé soit choisi.

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 la petite icône placée à côté du bouton pointillé. Le style fixé est valable pour toutes les primitives avec cependant quelques exceptions:

- pas de trait pointillé pour les tracés au crayon, les B-splines et les splines sauf si vous sauvez en Postscript
- pas de pointillé pour les ovales ou partie d'ovale
- le réglage de l'épaisseur des traits ne fonctionne pas avec les ovales si l'on sauve sous format **L^AT_EX**. Pour que cela marche il est nécessaire de sauver le dessin en Postscript.

2.11 Modifier un style de trait

On peut modifier globalement 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 pointillé puis sélectionner dans le menu du bouton **Existant** (en-dessous du bouton pointillé) **pour tous**, **pour les associees**, **pour la primitive selectionnee** ou **pour la derniere primitive entree**. Le dernier choix du menu de ce bouton permet de régler l'épaisseur des traits. Cette commande permet de régler à l'aide d'un curseur l'épaisseur des traits épais et pointillé épais. Le curseur qui apparaît permet de fixer l'épaisseur des traits en 1/10 de mm jusqu'à 3mm.

La commande **pour les traits epais** dans le menu associé au bouton **Existant** permet de modifier à postériori l'épaisseur des traits des primitives tracées en trait épais.

*Modifier
le style d'une
primitive ex-
istante*

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 flèche en haut



Sélection

En cliquant dans cette icône on peut sélectionner une ou plusieurs primitives, la déplacer, et changer ces dimensions.

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 (la primitive sélectionnée le reste jusqu'à la sélection suivante). Dans le cas contraire il y a une sonnerie et un message apparaît dans le mini-buffer.

Si les règles sont affichées les dimensions de la primitive sélectionnée s'affiche dans le panel d'information. Pour les primitives de type ligne, cercle, ellipse, rectangle, ovale après sélection apparaît une croix au centre de la primitive.

3.1.2 Sélection pour primitives recouvrantes

Si deux primitives p_1, p_2 se recouvrent en un ou plusieurs points il est parfois difficile de sélectionner une des deux (disons que p_1 est toujours sélectionnée et que p_2 ne l'est jamais).

Dans ce cas il suffit de maintenir le bouton L3 appuyé lors de la sélection pour que la sélection s'opère 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 de sélection sélective, voir page 40. Par ailleurs cette icône permet d'effectuer des commandes directes sans passer par les menus (voir page 85).

3.2 Association

3.2.1 Associer plusieurs primitives

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 trois manières.

- **Dans un rectangle** On peut sélectionner toutes les primitives qui sont incluses dans un rectangle en sélectionnant celui-ci. Pour cela il suffit de d'appuyer sur la touche L8 après avoir sélectionné le rectangle. Un '+' apparaît dans un coin du rectangle, ainsi que dans toutes les primitives associées. On peut alors couper, copier, translater toutes les primitives en bloc selon les mêmes procédés que pour une primitive unique. Attention, une erreur fréquente s'observe 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 Associes** du menu **EDITION**.
- **Primitives désignées** On peut associer directement une primitive en restant dans le mode sélection mais en cliquant avec le bouton de droite près de sa frontière.
- **Primitives du même type** La commande **Association Multiple** du menu **EDITION**, sous-choix de **Associer-Dissocier** permet d'associer ou de dissocier sélectivement des primitives du même type (exemple: tous les textes). Un bouton de cycle apparaît qui permet de sélectionner quel type de primitive on va associer ou dissocier. Les choix possibles sont: les lignes, les textes, les rectangles, les ovales, les ellipses, les cercles, les polygones.

Dissocier

Pour dissocier toutes les primitives associées il suffit de choisir **Dissocier Tout** sous-choix de **Associer-Dissocier** dans le menu **EDITION**. En mode "Selection" on peut le faire en tapant "d" dans le canevas.

Pour en dissocier une il faut la sélectionner et choisir **Dissocier la Selectionnee** dans le menu **EDITION**.

Chapitre 4

Manipulation de primitives

On va décrire ici quelques manipulations de base sur les primitives. On pourra aussi consulter le chapitre Menus.

4.1 Translater une primitive

Translater

-petit déplacement: 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). La primitive suit le déplacement de la souris (figure 4.1a).

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, ce qui va prendre du temps. Il est donc plus raisonnable dans ce cas d'opter pour la méthode des grands déplacements.

-grand déplacement: sélectionner un point de la primitive, relacher. Appuyer sur la touche L9 puis déplacer la souris à l'endroit où l'on veut placer le point sélectionné en maintenant L9 appuyée. Appuyer sur un bouton de la souris, la figure se déplace correctement (figure 4.1b).

-déplacement fin: Déplacer finement (de quelques pixels) une primitive à la souris est délicat. On peut alors utiliser les touches fonctions R1,R2 qui déplace la primitive d'un pixel respectivement à gauche et à droite ou R4,R5 qui font la même chose vers le haut et le bas de la page.

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

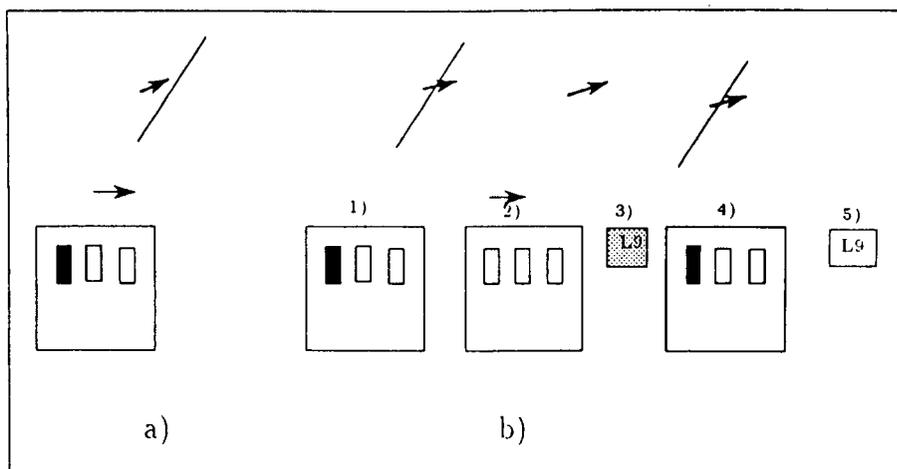


Figure 4.1: Comment traduire une primitive
 a) petit déplacement
 b) grand déplacement

l'englobe. Donc si on la déplace, seul le contour extérieur se translate. Pour la traduire 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 sélectionner une primitive, relâcher le bouton de la souris. Appuyer sur la touche L10 en permanence.

Modifier la taille

4.2.1 Cas des rectangles, cercles, lignes

Une étoile apparaît sur la primitive. Mettre le curseur en ce point et appuyer (le bouton L10 doit être maintenu appuyé en permanence). La primitive change de forme jusqu'à ce que l'on relâche le bouton. Pour les lignes on peut ainsi faire tourner le segment autour de son origine.

4.2.2 Cas des ellipses, ovales

Pour une ellipse ou un ovale en maintenant L10 appuyé on peut modifier la longueur du petit ou du grand axe de la primitive.

4.2.3 Cas des textes

Pour les primitives de texte la modification est d'un autre type: elle permet d'éditer le texte dans le mini-buffer emacs (voir page 29).

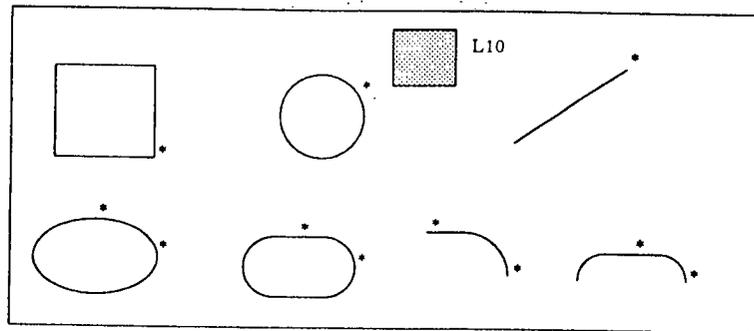


Figure 4.2: Modification de taille des primitives de dessins, action de L10

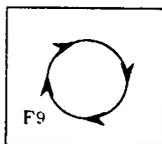
4.2.4 Cas des polygones

*Changer
la taille d'un
polygone*

Pour les polygones deux cas se présentent. Soit le point cliqué est proche d'un sommet, auquel cas ce sommet 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 sommet du polygone).

4.3 Rotation

Les rotations



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. 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 "+".

Une action sur le bouton de gauche de la souris fait tourner la figure de 15 degré, celui du milieu de 5 degré, celui de droite de 1 degré dans le sens indiqué par le bouton à droite de l'icône. Pour changer de sens il suffit de cliquer sur ce bouton.

Pour des rotations plus globales on peut utiliser un curseur. Pour cela, après avoir sélectionné l'icône de rotation et placé la souris dans le canevas, tapez sur la touche L8. Un curseur apparaît permettant d'ajuster l'angle de rotation. Le bouton FIN permet de faire disparaître cette fenêtre.

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 le menu **Detruire** avec comme sous-choix **Detruire la primitive selectionnee**. On peut aussi effectuer plus rapidement cette manipulation en conservant la souris dans le canevas et en tapant sur la touche fonction R13.

Detruire les primitives

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 les sous-choix **Afficher le Presse-Papier** du menu **Presse-Papier** dans le menu **EDITION**.

Le presse-papier

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**. Un raccourci pour coller après sélection est d'appuyer sur R7. 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 Redémarrer

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

Redémarrer

4.4.4 Détruire tout sauf les associées

La commande **Detruire tout sauf les Associees** du menu **Detruire** dans le menu **EDITION** permet de détruire toutes les primitives qui ne sont pas associées.

4.4.5 Détruire tout sauf la sélectionnée

La commande **Detruire tout sauf la Selectionnee** du menu **Detruire** dans le menu **EDITION** permet de détruire toutes les primitives sauf la primitive sélectionnée.

4.5 Correction de texte

Pour faciliter la correction de texte il existe une méthode simple basée sur un éditeur de type emacs. A chaque entrée de texte il se passe la chose suivante:

- on tape le texte complet que l'on termine en déplaçant la souris.
- le texte courant apparaît dans le mini-buffer emacs.

Un & remplace le retour chariot dans emacs!

Un retour chariot est indiqué dans le mini-buffer par un &. On peut rajouter des lignes au texte en tapant ce caractère dans le texte mais il ne faut pas en placer deux consécutivement. On peut alors se déplacer dans le mini-buffer emacs avec les commandes emacs classiques (figure 4.3):

- 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

emacs: █	qwerty	CTRLa
emacs:	qwerty █	CTRLe
emacs:	qwerty █ y	CTRLb
emacs:	qwerty █	CTRLf
emacs:	qwerty █	CTRLd
emacs:	qwerty █	CTRLt

Figure 4.3: L'action des diverses commandes emacs sur un texte

On peut donc rajouter des caractères, en détruire de manière commode. La fenêtre emacs contient toujours le dernier texte entré. Pour faire apparaître dans le mini-buffer 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).

4.6 Copier une primitive

Copier une primitive

Pour copier une primitive sélectionnée il faut choisir le sous-choix copier du menu **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

Copier avec une homothétie

Ce sous-choix de la commande **Copier** du menu **EDITION** permet de copier la primitive sélectionnée en exerçant 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.

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

4.8 Les miroirs, pivot et le zoom

Ces trois commandes permettent la manipulation de primitives et sont décrites précisément dans le chapitre consacré aux menus (menu **DISPOSITION** et menu **PAGE**). Définissons simplement leurs actions.

*Pivot, miroir,
zoom*

- **miroirs**: on va créer une primitive qui sera semblable à la primitive sélectionnée vu dans un miroir que l'on peut placer à côté, au dessus ou en dessous de la primitive.
- **pivot**: pour faire pivoter la primitive sélectionnée.
- **zoom**: pour agrandir ou réduire le dessin, ou une partie du dessin selon un axe ou bien dans les deux dimensions.

4.9 Couper une primitive

Pour créer une coupure sur une primitive existante il suffit de choisir dans le menu de l'icône des utilitaires l'icône avec les 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 du milieu.

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 de l'icône des utilitaires l'icône avec la loupe. Un carré apparaît dans le panel. Placer la souris à l'endroit où l'on désire obtenir la loupe et déplacer légèrement la souris en maintenant le bouton de gauche appuyé. La vision de la zone apparaît dans le carré. Si l'on désire obtenir ce grossissement sur une zone plus importante il suffit d'agrandir le carré comme pour une fenêtre normal. Il est possible dans ce mode d'utiliser les touches R1, R2, R4, R5 pour translater les primitives d'un pixel: la fenêtre affiche alors le nouveau résultat.

JPdraw et Postscript

5.1 Mélanger JPdraw et Postscript

JPdraw et
Postscript

Nous avons vu que JPdraw peut fonctionner du point de vue graphique sous deux modes. Le premier utilise les primitives graphiques standard de L^AT_EX et le second utilise un fichier d'extension qui contient des nouvelles commandes L^AT_EX de dessin qui font elles mêmes appel à des commandes Postscript. L'utilisation de ces nouvelles primitives pour la création de dessins un peu complexes est, bien sûr, nécessaire mais L^AT_EX ne sait pas compiler des fichiers où les dessins contiennent un trop grand nombre de primitives.

Pour permettre la création de gros dessins on va utiliser une commande de L^AT_EX qui permet d'inclure directement du Postscript. Le choix fait dans JPdraw de conserver la définition des primitives à l'aide d'ordre L^AT_EX 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 L^AT_EX pour le texte inclus dans un dessin.

JPdraw
peut sauver le
dessin en
Postscript!

Revenons maintenant au problème des dessins trop complexes pour que L^AT_EX puisse les imprimer. La solution idéale serait de panacher du code généré pour L^AT_EX (pour les textes) et du Postscript (pour le dessin proprement dit). C'est justement ce que peut faire JPdraw. 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 (je reviendrais sur la manière de faire cette opération) et regardons ce qui se passe au niveau des fichiers créés. On voit que JPdraw génère alors trois fichiers `xx.ltex`, `xx.macdraw`, et `xx.ps`.

Examinons tout d'abord le fichier L^AT_EX. Il se présente sous la forme suivante:

```
\begin{picture}( 2.5, 1.9)( 2.0, 10.4)
\thinlines
\large
\put(2.09,12.22){toto}
\put( 2.030,10.382){\special{ps:plotfile xx.ps}}
\end{picture}
```

On reconnaît la première instruction qui est celle qui permet d'écrire du texte dans le dessin. La seconde est:

```
\put( 2.030,10.382){\special{ps:plotfile xx.ps}}
```

C'est l'ordre qui permettra d'incorporer le fichier Postscript `xx.ps` à cet endroit dans le fichier que l'on va imprimer. Ce fichier Postscript se présente sous la forme suivante:

```
%%BoundingBox 10 100 10 100
/Ellipse { gsave moveto setgray /savematrix matrix
currentmatrix def currentpoint translate rotate newpath
scale arc closepath savematrix setmatrix stroke grestore} def
/EllipseColor { gsave moveto setgray /savematrix matrix
currentmatrix def currentpoint translate rotate newpath
scale arc closepath savematrix setmatrix fill grestore} def
/Ligne { gsave moveto currentpoint translate 0 0 moveto
lineto currentpoint stroke moveto closepath grestore} def
/ArrowNoire {gsave 0 setgray moveto currentpoint
translate 0 0 moveto lineto lineto lineto lineto
currentpoint fill moveto closepath grestore} def
/RectangleNoir {gsave setgray moveto currentpoint
translate 0 0 moveto lineto lineto lineto lineto
currentpoint fill moveto closepath grestore} 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} def
/ClosePoly {currentpoint stroke moveto closepath grestore} def
gsave
  currentpoint translate
  0.3 setlinewidth
  [] 0 setdash
  150.846 67.819 0.000 0.000 Ligne
grestore
```

La première partie est utilisée pour définir un ensemble de procédure Postscript nécessaire par la suite. Le tracé de ligne est effectué à l'avant-dernière ligne du fichier. Pour les professionnels de Postscript notez que JPdraw indique dans le dessin une BoundingBox.

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.

5.2 La commande special

Cette manière d'inclure un fichier postscript est celle qui fonctionne pour le programme `dvi2ps` fonctionnant à l'INRIA. Toutefois si ce n'est pas la syntaxe utilisée par le programme que vous

*La commande
"special"*

utilisez pour convertir un fichier .dvi en Postscript vous pouvez modifier cette instruction (voir page 77). Examinons maintenant la manière de créer les fichiers dont nous venons de parler: ces commandes se trouvent dans le sous-menu **Sauver** du menu **FICHIER**.

5.3 Créer les fichiers L^AT_EX et Postscript

5.3.1 La commande Sauver le Dessin en Postscript et le Texte en Latex

Cette commande permet de créer les fichiers L^AT_EX et Postscript qui vont contenir la description du dessin. Le nom du fichier Postscript porte l'extension .ps.

5.3.2 La commande Sauver le Dessin en Postscript et le Texte en Latex Sous

La commande est identique à la précédente mais demande simplement le nom du fichier dans lequel on va sauver le dessin. Le nom ne doit pas comporter d'extension .macdraw ou .ps.

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 à la même place précédemment (par exemple une texture blanche dans une ellipse masquera complètement tout autre élément ou partie d'élément contenue dans l'ellipse). Il faut donc être conscient que dans Jpdraw on a choisi un ordre à-priori entre la définition des textes et l'inclusion du fichier Postscript: on écrit d'abord les textes puis le fichier Postscript. Ceci a pour conséquence que si le texte se trouve sur une primitive texturée à l'écran il sera masqué et non-visible. Pour réparer cette erreur il suffit de placer l'ordre d'écriture du texte dans le fichier L^AT_EX après l'ordre d'inclusion du fichier Postscript.

Chapitre 6

Description détaillée des menus

Les menus

Les menus sont: EXTENSION, FICHIER, EDITION, STYLE, CARAC, PAGE, DISPOSITION, ANNULER DERNIER ORDRE 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. Parfois un sous-menu est disponible: ceci est indiqué par une flèche horizontale orientée vers la droite. Il suffit alors de translater la souris vers la droite pour faire apparaître le sous-menu. Pour passer d'un menu à un autre il est nécessaire de relâcher le bouton de la souris.

6.1 Le bouton EXTENSIONS

Le bouton EXTENSIONS permet de valider ou d'invalider les extensions \LaTeX . 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 . On se retrouve donc avec les contraintes correspondantes. \JPdraw gère alors les contraintes. Ainsi les lignes sont tracées avec une pente compatible avec \LaTeX car \JPdraw modifie l'extrémité de la ligne pour cela. La contrainte sur le diamètre maximum des cercles n'est cependant pas prise en compte.

Si on invalide les extensions et que l'on tente d'effectuer une opération qui les nécessite une fenêtre apparaît. On y indique que l'opération n'est pas permise et on demande si on valide les extensions.

Attention dans le mode fonction (voir page 51) 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 Redemarrer

-effacement du dessin et création d'un nouveau fichier. Si le fichier courant n'est pas à jour, c'est-à-dire que le dessin courant contient plus de primitives que le fichier, une icône apparaît (avec une tombe!) associée à des commandes: **continuer**, **enregistrer**, **annuler**. La première exécute **Redemarrer** sans mise à jour du fichier courant, **enregistrer** sauve les modifications sur le fichier courant et **annuler** annule la commande Redemarrer.

6.2.2 Charger

-permet de charger un fichier. Les seuls fichiers que l'on peut recharger sont ceux sous format **macdraw**. On peut toutefois omettre l'extension **.macdraw**. Mais la chaîne de caractères ne doit pas contenir d'autre extension.

6.2.3 Sauver

C'est un choix avec un sous-menu.

6.2.3.1 Sauver

-enregistrement du dessin sur le fichier courant.

6.2.3.2 Sauver Sous

Permet de sauvegarder un fichier sous un autre nom que le nom courant. Donner le nom du fichier suivi d'un retour chariot. Il ne faut pas donner au fichier un nom dont l'extension est **.macdraw**.

6.2.3.3 Sauver sous format raster

Permet de sauver un dessin sous forme d'image raster de SUN. On peut ensuite recharger ce dessin avec un **screenload**. Seul le dessin est sauvé. Choisir dans le menu FICHIER l'ordre **Sauver Sous FORMAT RASTER**. Une fenêtre texte apparaît demandant le nom du fichier raster. Placer la souris dans cette fenêtre (le petit triangle doit clignoter). Donnez ce nom et terminer soit en cliquant sur le petit carré à côté du OUI soit avec un retour chariot.

6.2.3.4 Sauver sous format inrimage

Permet de sauver un dessin sous format inrimage. En fait on recopie le "pixrect" du canevas de dessin ce qui implique que la taille de l'image est au plus de 1000x 1000. Une fenêtre apparaît demandant le nom du fichier inrimage et les caractéristiques de l'image. Par défaut on sauve l'image sous format 512x512. On peut choisir de sauver l'image sous format rectangulaire avec cadrage automatique en cliquant sur le carré à côté de **IMAGE ECRAN**. On peut aussi modifier la taille de l'image carré en cliquant sur le petit carré à côté de **IMAGE CARRE**. Trois curseurs apparaissent. Ils permettent de choisir la taille de son image. Puis on donne le nom du fichier.

Placer la souris dans cette fenêtre (le petit triangle doit clignoter). Donnez ce nom et terminer soit en cliquant sur le petit carré à côté du OUI soit avec un retour chariot.

Si la taille de l'image carré est supérieure à la taille du dessin on recopie le dessin tel quel. Pour connaître la dimension du dessin crée, en pixel, on peut utiliser l'instruction **Afficher les Regles Ecrans** dans le menu PAGE qui trace des règles en pixel sur les bords du canevas ou on peut demander **Taille du Dessin** dans ce même menu qui donne les dimensions en cm et en pixel du dessin courant. On peut aussi ajuster la taille de l'image en utilisant le zoom.

Attention les règles font partie du canevas, il est donc prudent de les effacer après avoir déterminé la taille du dessin (**Masquer les Regles Ecrans** dans le menu PAGE).

Si la taille de l'image carré est inférieure à la taille du dessin on procède à une réduction temporaire du dessin courant de façon à ce que celui-ci rentre dans l'image (une légère marge est laissée).

Pour pouvoir utiliser cette facilité il est nécessaire d'avoir accès à la commande "create" d'INRIMAGE.

6.2.3.5 Sauver le Dessin en Postscript et le Texte en Latex

Sauve le texte dans le fichier \LaTeX courant et le dessin au format postscript dans un fichier d'extension .ps. Voir page 32

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

Idem que précédemment mais on vous demandera d'abord le nom d'un fichier \LaTeX . Pour sauver donnez le nom du fichier suivi d'un retour chariot.

6.2.4 Impression Ecran

-imprime une copie de l'écran (attention commande fragile qui dépend beaucoup de votre système, voir la partie correspondante dans la section installation de JPdraw).

6.2.5 Verifier

On a le choix entre **Verification Latex** et **Verification LaTeX avec Postscript**. Dans le premier cas on le dessin est crée en \LaTeX et dans le second en Postscript. Cette commande crée un fichier `bidon.tex` contenant le dessin courant et qui est directement compilable par \LaTeX . Le fichier postscript généré s'appelle `xxx.ps`.

6.2.6 Album

-Permet de rajouter au dessin courant un dessin défini dans un autre fichier .macdraw. La commande demande le nom du fichier .macdraw. L'extension .macdraw n'est nécessaire que si une autre extension est présente dans le nom du fichier. Les primitives de l'album sont associées pour que l'on puisse les déplacer facilement.

6.3 Le Menu EDITION

Détruire

6.3.1 Détruire

Ce sous-menu gère la destruction des primitives.

6.3.1.1 Détruire la Primitive Sélectionnée

Détruit la primitive sélectionnée. Un raccourci possible est de sélectionner puis d'appuyer sur la touche R13.

6.3.1.2 Détruire Tout sauf les Associées

Détruit toutes les primitives du dessin sauf les primitives associées.

6.3.1.3 Détruire Tout sauf la Sélectionnée

Puis **Détruire Tout sauf la Sélectionnée** qui permet de détruire toutes les primitives du dessin sauf la primitive sélectionnée.

6.3.2 Eliminer les Primitives cachées

Permet de placer dans le presse-papier les primitives recouvertes par une primitive texturée en niveau de gris (donc effectivement cachées dans le dessin papier). Cette commande est très lente.

Copier

6.3.3 Copier

6.3.3.1 Copier la primitive sélectionnée

Copie la primitive sélectionnée. Un raccourci après la sélection est d'appuyer sur R6. La primitive copiée est strictement identique à la primitive sélectionnée.

6.3.3.2 Copier avec Homothétie

Permet de recopier la primitive sélectionnée avec un facteur d'homothétie, réglé par un curseur.

Le presse-papier

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

Rajoute au dessin une primitive sélectionnée dans le presse-papier. 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 **Selection** on peut réaliser cette opération en tapant un "a" dans le canevas.

6.3.4.3 Masquer le Presse-Papier

Pour revenir au dessin courant après avoir affiché le presse-papier. Si l'on est en mode **Selection** on peut réaliser cette opération en tapant un "m" dans le canevas.

6.3.4.4 Vider le Presse-Papier

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. Le nombre de primitives restant à éliminer est indiqué dans le mini-buffer. Attention, après cette commande la gomme n'agit plus sur les textures faites avec des lignes.

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

*Associer et
Dissocier*

6.3.5.1 Associer Tout

Associe toutes les primitives du dessin courant. Si l'on est en mode **Selection** on peut réaliser cette opération en tapant un "A" dans le canevas.

6.3.5.2 Associer les primitives du même type

Permet d'associer ou de dissocier sélectivement des primitives du même type (exemple: tous les textes).

6.3.5.3 Dissocier Tout

Dissocie les primitives associées et redessine le dessin complet. Si l'on est en mode **Selection** on peut réaliser cette opération en tapant un "D" dans le canevas.

6.3.5.4 Dissocier la Primitive Selectionnee

Dissocie la primitive sélectionnée de l'ensemble des primitives associées. Si l'on est en mode **Selection** on peut réaliser cette opération en tapant un "d" dans le canevas.

6.3.6 Selection Selective

Cette commande du menu **EDITION** permet de sélectionner par la suite uniquement des primitives du même type. Après déclenchement de cette commande apparaît un panel qui permet de choisir le type de primitive que l'on va sélectionner. Un message apparaît en permanence dans le coin supérieur gauche de l'écran 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 le panel de choix. Ce mode présente un intérêt si le dessin compte un grand nombre de primitives.

6.3.7 Changer la valeur de Unitlength

Nous avons vu qu'en standard une page JPdraw avait une taille de 14.5cm. Cette commande permet de modifier cette valeur et de faire ainsi un dessin sur une page complète. La taille du dessin est simplement multiplié (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. Un des reproches fait à JPdraw, utilisé dans le mode texte en \LaTeX et dessin en Postscript, était de ne pas pouvoir modifier la taille du dessin en changeant simplement la valeur de `unitlength`. A partir de la version 3.2 c'est maintenant possible en utilisant une macro des extensions `\setunitlength`. Par exemple pour grossir de 1.5 fois un dessin il suffit de remplacer dans le fichier \LaTeX les instructions:

```
\setlength{\unitlength}{ 1.0cm}
```

par

```
\setunitlength{ 1.5cm}
```

Attention le facteur de grossissement est relatif c'est-à-dire que l'argument est directement appliqué comme facteur d'échelle au Postscript. Les conséquences en sont que si vous avez enregistré votre dessin avec un `unitlength` différent de 1 il faudra opérer en deux étapes. Par exemple supposez que vous ayez enregistré un dessin avec un `unitlength` de 3cm et que vous vouliez le réduire en ramenant le `unitlength` à 1.5cm (donc réduire le dessin de 2 fois). Si vous appliquez directement le `setunitlength` avec 1.5cm vous aurez des textes convenables mais un dessin qui sera 3 fois trop gros (l'échelle en est de 2x1.5). Pour s'en sortir il faudra écrire

```
\setunitlength{ 0.5cm}
\setlength{\unitlength}{ 1.5cm}
```

La première instruction permet de diminuer de 2 la taille du dessin et la seconde fixe la bonne valeur de `unitlength` pour les textes. Si vous avez un doute sur la valeur avec laquelle vous avez enregistré le postscript la valeur est placée en commentaire dans le fichier Postscript.

6.3.8 Formule

Si l'on utilise cette commande on va voir apparaître un panel où l'on demande une formule, une valeur de départ, une valeur de fin et un pas. Ce mode permet de tracer à l'écran une fonction du type $y=f(x)$. La formule doit se donner sous cette forme: par exemple $y = \sin(x)$, $y =$

$0.3 * \sin(3 * (\log(x) + 1) - 2)$ sont des formules valides. **X0** est la valeur de départ de **x**, **X1** la valeur maximum de **x** et **Pas** sert à calculer la valeur de la fonction au différents points **X0+Pas**, **X0+ 2*Pas** etc.... Les opérations reconnues dans ce mode sont **x**, **/**, **+**, **-**, ****** et les fonctions connues sont *exp*, *log*, *log10*, *sin*, *cos*, *tan*, *atan*, *asin*, *acos*, *sinh*, *cosh*, *tanh*, *sqrt*. De plus la constante π est connue. La position du 1er point sert à définir la position initiale du point définie par $y_0=f(x_0)$. Si l'on ne définit que la formule et la valeur de **X0** la valeur de la fonction s'affiche dans le panel d'information: on peut donc se servir de ce mode comme une calculatrice 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. Dans ce panel apparait 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

Aligner

Cette commande est très puissante. Pour aligner des primitives par rapport à une primitive fixe il suffit d'associer les primitives voulues en commençant par la primitive fixe. Puis exécuter la commande **Aligner**. Un panel apparait en bas des icônes. Les boutons qui apparaissent indiquent quelle partie des primitives on va aligner et le menu associé aux boutons permet de stipuler quelle partie de la primitive fixe est utilisée comme référence.

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.

6.4 Le Menu STYLE

Le style des textes

Sélectionne le style des textes. Choix: standard, gras, italique et souligné. Le style courant est indiqué en grisé.

6.5 Le Menu CARACT.

La taille des caractères

6.5.1 10pt...24pt

Sélectionne la taille des caractères (10pt par défaut, 24, 18, 14, 12 ou 10). La taille courante est indiquée en grisé.

6.5.2 Texte existant

Permet de changer la taille des textes déjà entrés de quatre manières.

6.5.2.1 Pour tous

On choisit la nouvelle taille et/ou le style désirés puis on sélectionne **pour tous** dans ce menu. Tous les textes du dessin se dessinent alors dans le style et la taille choisis.

6.5.2.2 Pour les Associees

On associe des primitives de texte puis on sélectionne le style et/ou la taille des caractères. Enfin on sélectionne **Pour les Associees** dans ce menu. Seules les primitives associées changent de style et/ou de taille.

6.5.2.3 Pour la Primitive Selectionnee

On sélectionne une primitive de texte et on choisit **Pour la Primitive Selectionnee** dans ce menu.

6.5.2.4 Pour la Derniere Primitive Entree

Pour la dernière primitive de texte entrée en choisissant pour **Pour la Derniere Primitive Entree** dans ce menu.

6.6 Le Menu PAGE

6.6.1 Redessiner Tout

Redessine le dessin complet. Permet aussi de faire disparaître les fenêtres d'erreur et de message. Le raccourci pour cette commande est F5.

6.6.2 Regles

Permet d'afficher ou de masquer dans le canevas des règles graduées en centimètre ou en pixel selon le choix dans le sous-menu. La même commande permet de masquer ces règles. Dans ce mode un texte dans le mini-buffer indique à tout moment les coordonnées de la souris en centimètre ainsi que les dimensions des primitives créées. Si l'on est en mode **Selection** on peut réaliser cette opération en tapant un "r" dans le canevas (cette opération affiche et masque les règles alternativement).

6.6.3 Taille

6.6.3.1 Taille du Dessin

Affiche dans le mini-buffer la position du dessin dans la page, les dimensions actuelles en cm et les dimensions en pixel.

6.6.3.2 Taille Reelle, 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 Reelle** permet d'avoir sur l'écran la page complète. **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.

6.6.5 Zoom

Zoom

Permet de diminuer ou d'agrandir un dessin. Un curseur apparaît avec laquelle on fixe le rapport d'agrandissement (Figure 6.1). 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 agrandit seulement le cercle.

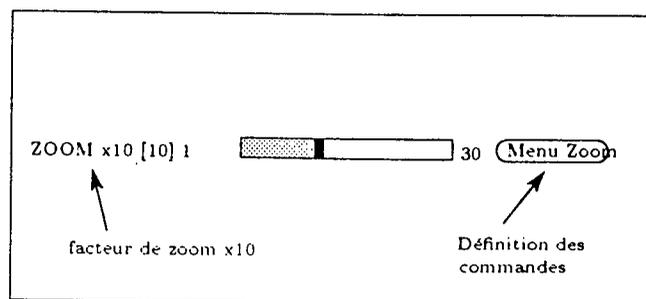


Figure 6.1: la fenêtre de zoom

Le zoom est équivalent à un changement de la taille de la feuille de papier équivalent au canevas. Normalement le facteur de zoom maximum est ajustée pour que le dessin ne puisse pas sortir du canevas. On peut obtenir un zoom complet de deux autres manières, soit en restant dans Jpdraw, soit en éditant le fichier \LaTeX (voir page 40). Après un zoom le dessin est centré.

6.6.6 Configuration

Le temps entre deux sauvegardes peut être modifié à l'aide de la commande du sous-menu. Ce temps est automatiquement augmenté en fonction du nombre de primitives.

6.6.7 Centrer

Centrer le dessin

On trouve dans le sous-menu les commandes **Centrer Tout** et **Centrer les Associees**. Ces commandes ont pour effet de placer le dessin si possible au centre du canevas visible sans bouger les scrollbars 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 **Selection** on peut réaliser cette opération en tapant un "c" dans le canevas.

Les plans

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 sont effectives seulement pour les primitives texturées en niveau de gris. Pour cela il suffit de sélectionner la primitive. Elle apparait 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).

Pivot

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 degré.

Les miroirs

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. 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). Normalement les primitives associées subissent toutes le miroir.

La réglure

6.7.4 Réglure

Le choix **Reglure** permet de placer un quadrillage invisible dans la fenêtre de dessin. Lorsqu'une primitive est créée, par exemple une ligne, le point de départ ne sera pas celui que vous avez donné mais le noeud du quadrillage le plus proche. Il en sera de même pour l'extrémité de votre segment. C'est utile lorsque vous voulez créer des dessins avec des éléments qui se coupent exactement. Par exemple un bloc diagramme avec des droites qui coupent exactement un carré. Par défaut la distance entre deux noeuds du quadrillage invisible est de 0.1mm.

6.7.4.1 Aligner la Selectionnee sur la Reglure

Permet de redisposer la primitive sélectionnée selon la réglure magnétique.

6.7.4.2 Aligner les Associees sur la Reglure

Permet de redisposer toutes les primitives associées selon la réglure magnétique.

6.7.4.3 Aligner Tout sur la Reglure

Permet de redisposer toutes les primitives déjà créés selon la réglure magnétique.

6.7.4.4 Modifier la reglure

Permettent de modifier les distances entre lignes horizontales et verticales du quadrillage invisible en indiquant à l'aide de sliders les nouvelles valeurs, soit en pixel, soit en mm (dans l'ordre x,y).

6.7.4.5 Ordre Inverse

Cette commande du menu disposition dessine les primitives dans l'ordre inverse de leur apparition à l'écran donc les primitives au second plan seront dessinées au premier plan.

6.8 ANNULER DERNIER ORDRE

Ce bouton permet d'annuler le dernier ordre entré. Tout les ordres ne sont pas annulables.

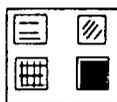
6.9 Le Menu BIB

Ce bouton avec menu permet d'exécuter un des programmes de la directory des bibliothèques (voir le chapitre Configurateur et le chapitre Programmation). Pour cela il suffit de choisir un des programmes du menu.

Chapitre 7

La texture des primitives

7.1 Texture



Les textures

Permet de texturer des rectangles, des ellipses, des cercles (ou des arcs de cercle) et des polygones. A ce sujet noter qu'une ligne polygonale non fermée n'est en fait qu'un polygone dont la dernière ligne n'est pas visible à l'écran. On peut donc la texturer de la même manière. On ne peut pas texturer les ovales (parce que l'arrondi des ovales n'est, en général, pas connu). La texture des arcs de cercles est une texture qui s'appuie sur la corde de l'arc. Les primitives associées sont toutes texturées.

7.1.1 Comment texturer une primitive

Pour texturer voilà comment procéder: sélectionner la primitive à texturer. Choisir le type de texture voulu dans le menu de l'icône texture. Pour cela on peut utiliser soit le menu (bouton de droite) soit cliquer avec le bouton de gauche sur l'icône: les différentes possibilités de texture apparaissent successivement. Une fois la texture choisie cliquer sur le bouton OUI à côté de l'icône de texture (ou taper 'o' dans le canevas).

Remarque Une fois que vous avez sélectionné une primitive puis sa texture et que vous avez cliqué sur le OUI vous n'avez plus à repasser par l'icône de sélection pour texturer une autre primitive. Il suffit de cliquer sur la frontière de la primitive voulue puis de choisir la texture.

7.1.2 Textures faites de lignes

Certaines textures sont faites à partir de lignes, qui sont des primitives de JPdraw (donc ces textures risquent d'être lourdes en nombre de primitives).

7.1.2.1 Type de texture

Dans JPdraw existent des textures faites de lignes horizontales, verticales, les deux ou bien de lignes inclinées. En standard il existe quatre types de texture avec des lignes horizontales, verticales, une grille où les lignes sont distantes de 0.5cm et enfin une texture avec ligne oblique avec par défaut un angle de 45 degré. 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.

7.1.2.2 Créer votre texture

Vous pouvez régler la distance entre vos lignes de texture ainsi que leur pente. Pour cela sélectionner **vo**tre texture dans le menu associé à l'icône des textures. Une fenêtre apparaît où l'on peut fixer les caractéristiques d'une texture (Figure 7.1).

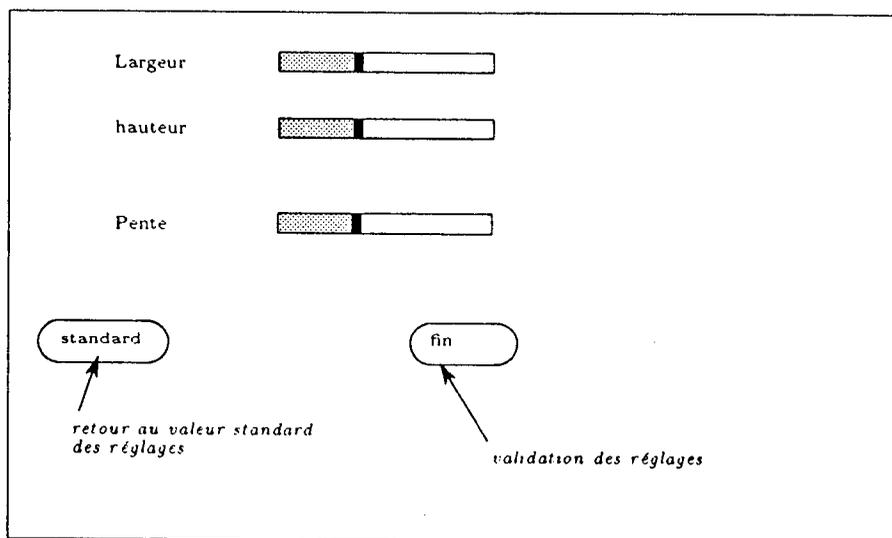


Figure 7.1: Le panel permettant le réglage des paramètres d'une texture

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. Une fois votre texture fixée appuyer sur **FIN**. Pour revenir au dimension de 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 le 5eme choix du menu de l'icone de texture on peut indiquer une texture à niveau de gris ou en couleur pour la primitive sélectionnée. Après avoir choisi l'icone et cliqué sur le OUI apparaît 4 sliders. Dans le premier on peut fixer le niveau de gris de la texture entre 1 (noire) et 15 (blanche). Le bouton **BORDURE NOIRE** permet de supprimer ou conserver la bordure de la primitive. Le bouton **Valider: niveau de gris, Fin** permet de fixer la texture. Les 3 autres sliders 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 **Valider: couleur** ce qui fait apparaître la couleur. Attention sur un SUN monochrome en place de la couleur la texture est noire et pour les textures en couleur la création de couleur a un effet de bord: le panel devient noir.

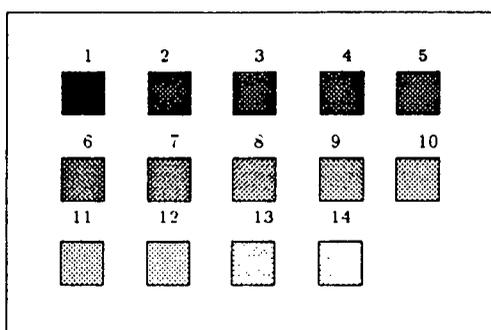


Figure 7.2: Les niveaux de gris

La texture s'affiche à l'écran avec toutefois une particularité de la texture blanche qui est qu'elle est en fait d'un gris léger. Pour les arcs la texture remplit l'espace entre l'arc et sa corde.

7.1.3.2 Les masques

Les masques

La texture blanche comporte une propriété intéressante: toute primitive définie préalablement et dont une partie est recouverte par une primitive de texture blanche verra la partie en question non tracée. 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 sans supprimer la bordure du rectangle qui nous a servi à masquer une partie de l'ellipse.

7.1.4 La gomme

Pour faire disparaître une texture dans une primitive il suffit de sélectionner cette primitive et de choisir l'icone gomme dans l'icone de texture.

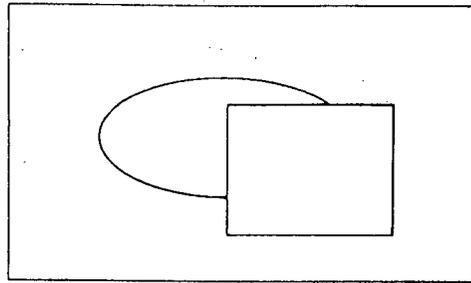


Figure 7.3: L'utilisation d'un masque pour créer un arc d'ellipse

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 et les polygones. Cela ne peut pas marcher pour les ovales car on ne connaît pas la représentation exacte de ces arrondis en \LaTeX .

Des trous dans une texture? Facile..

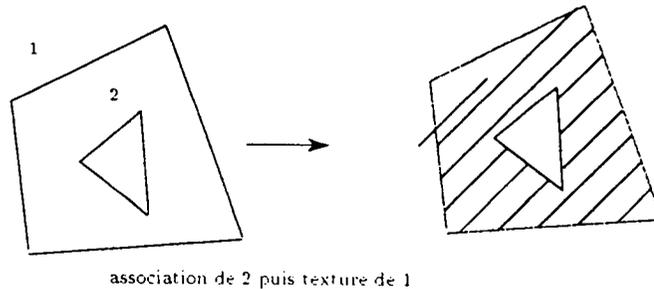


Figure 7.4: Texture avec des trous

7.1.6 Remarques sur les textures

Attention: La texture n'est pas associée à la primitive qui la contient. C'est un choix qui permet par exemple de générer des textures sans bord en créant une primitive, en la texturant puis en supprimant la primitive englobante. Cela pose des problèmes pour les déplacements de ce type de primitive. Il y a toutefois des solutions comme l'association (voir page 25).

Attention: si on a fait tourner une figure la texture associée ne bouge pas.

7.1.6.1 Restriction sur les textures des tracés crayon

Une texture grise sur un tracé crayon apparaît à l'écran comme recouvrant complètement le tracé (frontière = le tracé et la ligne reliant les points extrêmes du tracé). Sur le papier on n'obtiendra cela que si l'on sauve le dessin en Postscript.

*Une texture
pour tous!*

7.1.7 Rendre une texture permanente

Il est parfois intéressant de définir une texture qui sera utilisée systématiquement pour toutes les primitives qui vont être créées par la suite. Pour cela il suffit tout d'abord de texturer une primitive avec la texture voulue puis de la définir comme permanente à l'aide du bouton au-dessus de l'icône des textures. C'est un bouton à menu qui permet de passer soit à **Temporaire** (la texture ne s'applique qu'à la primitive sélectionnée), **Permanente** (les primitives créées par la suite auront la texture définie) ou **Définir** qui permet de définir une nouvelle texture que l'on veut rendre permanente.

Chapitre 8

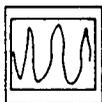
Les fonctions utilisateurs

8.1 Fonction

8.1.1 Généralités

Dans les deux chapitres suivants nous allons décrire comment il est possible de faire générer un dessin par JPdraw à partir de données externes. En effet un gros problème auquel on se trouve confronté avec les outils de dessin actuels est que parfois un dessin ne peut pas être effectué à la souris. Un exemple évident est le tracé de fonctions issues de résultats expérimentaux. Certains programmes proposent des sorties en Postscript mais cela pose des problèmes: le premier est que l'édition du résultat est impossible et le second est qu'en général les textes du dessin ne sont pas dans la même typographie que celle utilisée dans le texte. Nous proposons deux outils permettant de surmonter ces problèmes. Le premier va être la création d'un fichier de données une *fonctions utilisateur*, qui sera interprétable par JPdraw et dans lequel se trouve décrit numériquement des primitives graphiques. Le second outil est un langage de programmation simple dans lequel des ordres de créations de primitives sont possibles.

8.1.2 Les fonctions utilisateurs



La fonction associée à cette icône est de permettre de tracer des fonctions ou toutes autres figures définies par un utilisateur dans un fichier en utilisant un méta-langage. En fait c'est un interface puissant entre un programme utilisateur et JPdraw. On a ainsi pu connecter avec JPdraw un programme de CAO et un programme de segmentation d'image (dans ce dernier cas cela marche dans les deux sens puisque l'on peut générer une image à partir d'un fichier

JPdraw, voir page 36). Il est facile de prévoir vos travaux de recherche de façon à ce que les résultats de vos programmes soient directement sous cette forme. Ainsi si vous avez un programme qui calcule les valeurs d'une fonction vous pourrez inclure directement un graphe de cette fonction dans vos futurs articles de manière confortable sans passer par de fastidieuses étapes de recherche de papiers et de figures.

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.

8.1.3 Comment charger un fichier utilisateur

On définit d'abord un rectangle qui va contenir le dessin défini dans le fichier utilisateur et on le sélectionne. A ce stade 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 gère les changements d'échelle, de façon à ce que le dessin occupe l'intégralité du rectangle. Faire apparaître le menu de l'icône fonction en cliquant avec le bouton de droite. Choisir la première icône. Relâcher le bouton de la souris. Une fenêtre de texte apparaît où l'on donne le nom du fichier qui contient les valeurs à traiter. Terminer par un retour chariot. Si le fichier n'existe pas il y a une sonnerie et la fenêtre fichier disparaît.

Si vous n'avez pas défini le rectangle qui va contenir votre tracé de fonction il y a une double sonnerie. Si vous n'avez pas commis d'erreur de manipulation vous devriez voir alors apparaître votre dessin. Deux autres manières de procéder sont possibles (voir pages 81,70).

8.1.4 Les fichiers utilisateurs: le meta-language

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. Plus exactement nous verrons qu'à chaque description de primitives est lié une succession d'arguments fixée. Une **ligne** est en fait le nombre de ces arguments. Il est cependant sain de placer chacun de ces arguments sur une ligne séparée pour que le nombre d'argument corresponde effectivement au nombre de lignes du fichiers. Nous allons décrire les différents modes.

8.1.4.1 Le mode comment

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

Exemple

```
mode comment
commentaire de la fonction utilisateur:
#
```

8.1.4.2 Le mode connecte

On dispose d'un fichier de points qui décrit une fonction continue sous la forme de couple de coordonnées x,y de la fonction. Les points sont alors tracés et sont reliés par des segments. Il existe plusieurs types de connection pour ce mode:

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

Tracé d'une rampe commençant en (0,0), passant par (1,1) et horizontal a partir de ce point jusqu'à (1,2). Les deux fichiers utilisateur suivants sont équivalents.

```

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

```

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

8.1.4.3 Le mode deconnecte

L'exemple typique de l'utilisation de ce mode est le tracé de segments non connectés entre eux. On doit alors fournir un fichier dont chaque paire de points (décrit par ces coordonnées) constitue le segment à tracer. Le fichier doit alors se mettre sous la forme:

```

mode deconnecte
x1 y1
x2 y2
.. ..

```

et ainsi de suite. Les différentes formes de ce mode sont:

standard	vecteur	pointillé
deconnecte	deconnecte_vecteur	deconnecte_pointille
deconnecte_epais	deconnecte_vecteur_epais	deconnecte_pointille_epais

qui indique comment sont reliés les points décrivant les segments. Le cas précédent peut ainsi s'écrire:

```

mode deconnecte
0 0
1 1
1 1
2 1

```

8.1.4.4 Le mode cercle

Il permet de rajouter un cercle ou une ellipse dans le dessin. Pour cela il suffit de rajouter dans le fichier:

```
mode cercle
x,y du centre du cercle
rayon selon x, rayon selon y (égaux si c'est un cercle)
```

Les formes de ce mode sont:

standard	pointillé
cercle	cercle_pointille
cercle_epais	cercle_pointille_epais
cercle_noir	

Le cercle sera noirci en mode `cercle_noir`. Par exemple pour tracer un cercle de rayon 1 unité centré en 0,0.

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

On peut 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
```

8.1.4.5 Le mode arc

Permet la définition d'arc de cercle. La définition est:

```
mode arc
x,y du centre du cercle
rayon rayon
angle1 angle2
```

où *angle1* et *angle2* sont les angles en degré qui définissent l'arc. Ces angles sont compris entre 0 et 360 degrés et l'arc est défini comme partant de *angle1* et allant vers *angle2*. Les formes 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 commençant à 10 degrés et terminant à 200. On obtient alors l'arc de la figure 8.1.

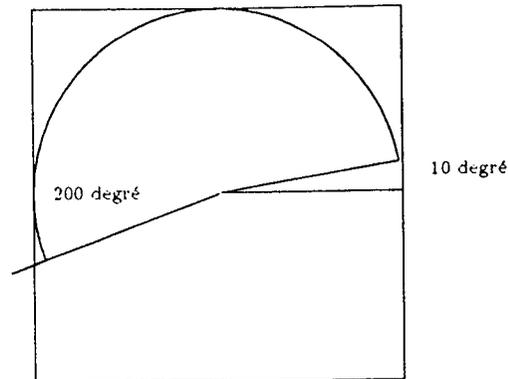


Figure 8.1: Un arc de cercle

8.1.4.6 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 /* position du texte */
type chaine /*chaine est la chaîne a placer*/

```

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. 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 est négatif.

Exemple

```

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

```

8.1.4.7 le mode rectangle

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

```
mode rectangle
x y /* origine du coin inferieur gauche du rectangle */
largeur hauteur
exemple
```

```
mode rectangle
```

```
0 0
```

```
1 2
```

On dispose de plus des modes:

standard	pointillé
rectangle	rectangle_pointille
rectangle_epais	
rectangle_noir	

8.1.4.8 Le mode polygone

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

```
mode polygone
nombre de point type
x1 y1
x2 y2
```

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	
	polygone pointillé épais	ligne polygonale pointillée épaisse	

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 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 1500x30points. Pour faciliter cette utilisation on a la possibilité en mode polygone de ne pas fixer le nombre de points.

Pour cela, après la ligne mode polygone il suffit de mettre le chiffre -1 comme nombre de points du polygone pour que JPdraw génère correctement les polygones. Par exemple si vous utilisez le fichier suivants:

```

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

```

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

Remarque: Cette facilité vous permet de tracer des dessins avec un grand nombre de point mais il est probable que dans ce cas L^AT_EX va refuser le tracé par manque de place mémoire. Vous devrez alors générer partiellement votre dessin en postscript comme il est indiqué page 37.

8.1.4.9 Mode bspline

Ce mode permet de créer des courbes de type bspline en indiquant les 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.

8.1.4.10 Mode spline

Syntaxe identique au mode bspline. Mais la courbe passe par les points de contrôle.

8.1.4.11 Le mode epaisseur_trait

Ce mode permet de définir l'épaisseur des traits pour les primitives qui le suivent. 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
```

8.1.4.12 Le mélange des modes

Il est possible de passer d'un mode à l'autre dans un fichier.

Exemple Tracé d'un tronçon de cône vue de dessus. Les commentaires ne font pas partie du fichier.

```
(arc d'ellipse)
-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
(trace des cotes du cone)
mode deconnecte
-1.000000 0.000000
-0.500000 4.000000
1.000000 0.000000
0.500000 4.000000
(trace de l'ellipse)
mode cercle
0 4.000000
0.500000 0.707107
mode texte
0 0
0 voici un tronçon de cone
```

8.1.5 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. Si JPdraw détecte des anomalies 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 fonction 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 fonction a été incorrectement défini. Le dessin ne sera pas créé.

Dans les deux cas on imprime à la console le numéro de la ligne incriminée (approximativement..).

8.1.6 Tracé d'abscisses et d'ordonnées

Pour tout dessin créé par une fonction utilisateur il est possible de faire apparaître automatiquement un quadrillage du rectangle et une impression des coordonnées (par défaut la distance entre les lignes du quadrillage est de 2 cm et les coordonnées ont 2 décimales) en sélectionnant la deuxième icône de l'icône fonction.

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. Dans ce cas une fenêtre apparaît avec des curseurs qui permettent d'ajuster la distance entre les lignes du grillage (soit en centimètres, soit en unité utilisateur) et aussi le format de sortie des abscisses et ordonnées qu'il faut donner sous la forme "n.m" où n et m sont des entiers. Une fois que le format de sortie a été donné terminer par un retour chariot. Cette opération doit être effectuée avant le tracé du grillage.

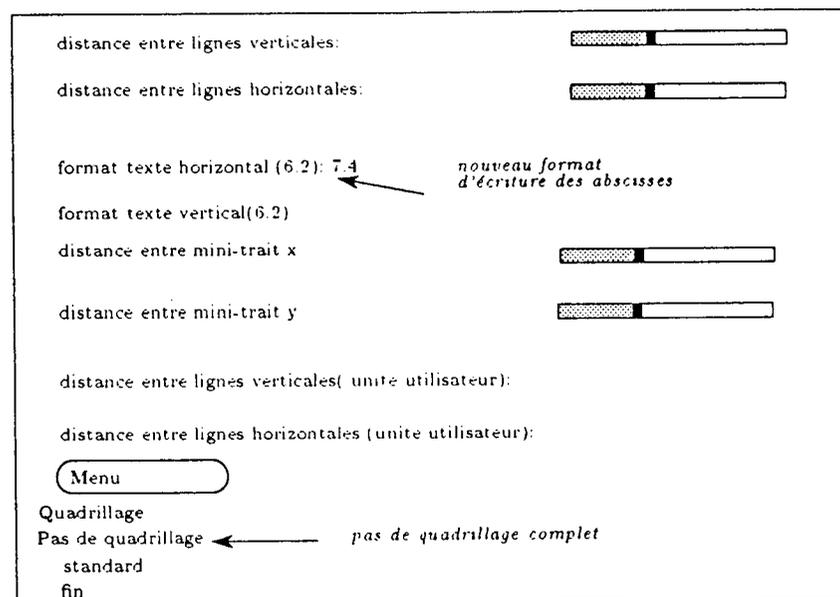


Figure 8.2: Le panel permettant d'ajuster un quadrillage

Dans cette fenêtre le choix dans le menu **pas de quadrillage** permet d'obtenir simplement les valeurs des abscisses et ordonnées avec une marque courte sur les bords du rectangle.

Les mini-traites sont simplement des petites graduations entre les graduations principales. Les curseurs des mini-traites permettent de régler la distance entre les mini-traites 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).

Il est possible de tracer plusieurs fonctions 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.

8.1.7 Echelle fixe

Il est possible de fixer les coordonnées maximum et minimum des points contenus dans le rectangle de la fonction. Pour cela, après avoir fait apparaître le rectangle et cliquer dans le premier item du menu de l'icône fonction, cliquer dans le bouton de choix **echelle fixe**. Une fenêtre texte apparaît où vous devez introduire les maxima et minima des coordonnées des points contenus dans le rectangle, terminés par un retour chariot. Par exemple si vous avez dans votre fichier utilisateur des segments issus d'une image 512x512 vous introduisez les max et min suivant: 0 512 0 512. Puis vous donnez le nom de votre fonction. Une autre manière de procéder est indiquée page 81.

8.1.8 La couleur et les niveaux de gris

8.1.8.1 Couleur

La notion de couleur pour les primitives existe dans le méta-langage qui permet de décrire des dessins pour JPdraw. 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**. Les valeurs doivent être comprises entre 0 et 255. Pour une primitive avec intérieur la bordure est alors tracée en noire. Pour ne pas avoir de bordure dans ce cas il faut utiliser:

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

Enfin pour tracer des polygones ou des cercles (ou ellipses et arcs) d'une certaine couleur sans remplissage utilisez:

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

8.1.8.2 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). Si ce nombre est négatif la primitive n'aura pas de bordure.

Exemple

```
mode connecte
0 0
0 9
```

```
9 9
9 0
0 0
modegris90 polygone
-1 0
5 5
3 4
4 2
mode90 rectangle
0 0
1 2
mode-95 rectangle
2 2
1 2
mode80 cercle
4 4
1 1
mode-80 cercle
5 5
0.5 0.5
mode90 cercle
7 7
0.4 0.7
mode-90 cercle
8 8
0.4 0.7
```

8.2 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
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 rot.	-
cercle_pointille	x,y centre	petit.grand axe, angle rot.	-
cercle_epais	x,y centre	petit.grand axe, angle rot.	-
cercle_pointille_epais	x,y centre	petit.grand axe, angle rot.	-
cercle_noir	x,y centre	petit.grand axe, angle rot.	-
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 contr., type	x,y points contr.	-
spline	nb point contr., type	x,y points contr.	-
epaisseur_trait	épaisseur des traits	-	-

Chapitre 9

Programmation

9.1 Introduction

Le langage décrit ci-après permet la construction de dessins à partir d'un fichier. A la différence des fonctions utilisateurs la description des primitives est obtenu par des calculs à l'intérieur du programme. L'unité utilisée est le centimètre. Les programmes écrit dans ce langage sont exécuté en lançant `jpdraw` par:

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

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

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

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

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

comme par exemple

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

Certains de ces programmes pourront aussi être exécutés en les lançant à partir du menu du bouton **BIB**. Ce sont les programmes que l'on trouve dans la directory indiquée dans le configurateur et qui ont l'extension `.fi`. Seul les programmes non paramétrés sont exécutables.

9.2 Variables et expressions

9.2.1 Expressions élémentaires

Une expression élémentaire est une collection de nombre et d'opérateurs respectant les règles de la programmation en C.

Exemple:

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

Les opérateurs élémentaires reconnus sont:

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

Certaines fonction mathématiques sont aussi reconnues. Ce sont:

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

rint retourne la partie entière d'un flottant. Le nombre π est aussi connu, il s'écrit pi. On peut construire des expressions élémentaires à partir de l'ensemble des opérateurs.

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

9.2.2 Variables

Une variable commence par un % et est 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)
```

9.2.3 Valeur

On a

```
[valeur]= [expression elementaire] ou [variable]
```

9.2.4 Coordonnées

Une coordonnée est un tableau de deux flottants. On peut affecter un nom à une coordonnée qui commence par %v suivi d'un entier. Une coordonnée se définit par:

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

on peut donc écrire:

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

ou

```
%v1=(%10,%11)
```

On a donc:

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

ou

```
= %v[entier]
```

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

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

On peut aussi faire l'inverse:

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

9.3 Chaîne de caractères

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

9.4 Impression

```
print([chaîne de caractere])
print([chaîne de caractere],[chaîne de caractere])
print([chaîne de caractere],[variable],[chaîne de caractere])
print([chaîne de caractere],[coordonnées],[chaîne de caractere])
print([numero primitive])
```

Exemple

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

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

9.5 Style des traits

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

9.6 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 JpDraw) 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 JpDraw émet un message dans la fenêtre d'information en indiquant qu'il attend des points. On doit alors passer en mode **Selection** 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 tableaux de coordonnées `\%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 créées préalablement à la rencontre de cette instruction vont être répliqués.

9.7 Emission de message vers la console

Il est possible d'émettre des messages sur la 2eme fenêtre d'information de Jpdraw par:

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

Si vous utilisez l'instruction `input_mouse` il est nécessaire de placer l'instruction `info` avant cette instruction.

9.8 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

```
[numero primitive]=#[0-9]
```

Dans tout ce qui suit on peut toujours affecter un numéro de primitive à la primitive créée.

9.8.1 Texte

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

où valeur est la taille des caractères (10, 12, 14, 16, 18, 24).

9.8.2 Cercles, Ellipses

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

Dans le premier cas on donne la position du centre du cercle et son rayon. Dans le second trois points sur son périmètre. On peut récupérer le rayon d'un cercle dans une variable par:

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

Le dernier cas permet de définir une ellipse en donnant son centre, les dimensions de son petit et son grand axe.

9.8.3 Rectangle

```
rectangle from [coordonnees] to [coordonnees]
rectangle from [coordonnees] length [coordonnees]
```

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 hauteur et la largeur.

9.8.4 Ligne

```
line from [coordonnees] to [coordonnees]
line from [coordonnees], vector [coordonnees], length [valeur]
line from [coordonnees] [cote] perpendicular to [numero primitive], length
[valeur]
line from [coordonnees] parallel to [numero primitive], length [valeur]
```

Dans le premier cas on définit la position des points de départ et de fin. Dans le second on donne la position du point de départ, le vecteur directeur et la longueur de la ligne. Dans le troisième cas on donne le point de départ et la ligne sera perpendiculaire à la primitive donnée, du côté indiqué, de la longueur indiquée. Dans le quatrième cas la ligne sera parallèle à la primitive donnée, de la longueur indiquée. [cote] peut prendre les valeurs **left** ou **right**.

9.8.5 Courbes

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 [coordonnees]
```

comme par exemple:

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

Ainsi le programme:

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

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).

9.9 Manipulation des primitives

9.9.1 Destruction de primitives

```
destroy [numero primitive]
```

9.9.2 Association, dissociation

On peut associer ou dissocier toute les primitives par **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, commençant à 0.

9.9.3 Translation, rotation

Il est possible de traduire une primitive par:

```
translate \#[numero] [coordonnees]
```

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

```
rotate \#[numero] around [coordonnees] , [valeur]
```

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

9.9.4 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*.

9.9.5 Miroirs

Il est possible de faire des miroirs par les ordres:

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

9.9.6 Centrer

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

9.10 Programme paramétré

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

```
jdraw -F [nom du fichier programme] [[argument0,argument1]]
jdraw -F formule.f[0,2,toto]
```

9.10.1 Déclaration des arguments

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

```
prog([nombre d'argument])
```

Exemple

```
prog(4)
```

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 nombre incorrect est utilisé.

9.10.2 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, commençant à 0.

9.10.3 Unitlength

```
set unitlength [nombre]
```

Permet de modifier la valeur de unitlength.

9.10.4 Commentaires

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

9.10.5 Sauver et charger un fichier

```
load [chaîne de caractere,argument]
```

```
save [chaîne de caractere,argument]
```

Exemple:

```
save "toto.ltex"
```

```
save $0
```

```
load "tata"
```

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

```
load $0
```

```
set unitlength $2
```

```
save $1
```

```
quit
```

et que l'on lance avec

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

Le fichier toto 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 lancé en invisible et peut donc être effectué à partir d'un simple terminal.

9.10.6 Fin de programme

```
quit
```

9.11 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 JPdraw. Ainsi

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

va charger la fonction `toto` dans un rectangle de dimension 10,10.

9.12 Divers

9.12.1 Distance entre deux points

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

9.12.2 Centre d'une primitive

```
[coordonnees]=center [numero primitive]
```

9.12.3 Intersection de deux primitives

```
[coordonnees]=intersection [numero primitive] [numero primitive]
```

9.12.4 Coordonnées de départ et de fin d'une primitive

```
[coordonnees]=start [numero primitive]
```

```
[coordonnees]=end [numero primitive]
```

9.13 Exemples de programme

9.13.1 Exemple 1

```
%1=3
%2=4
%10=5
%11=6
line from (%1,%2) to (%10,%11)
%v0=(7,8)
%v1=(10,11)
print(%v1[0],%v1[1],"\n")
#1=line from %v0 to %v1
%v2= end #1
#2=line from %v2 , vector (9,1) , length 3
```

```
%v3= start #2
print(%v3)
#3=line from %v3 , vector (9,-1) , length 3
print(#3)
quit
```

9.13.2 Exemple 2

```
/*-----
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 point 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 */
```

72 JPdraw

```
%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 futur 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

```

Chapitre 10

X et Jpdraw

Il existe une version expérimentale de Jpdraw sous X. Elle ne dispose cependant pas de l'ensemble des fonctionnalités de la version Suntools en raison du nombre très importants d'erreurs que l'on peut trouver dans ce système de fenêtrage. On résume ici les différences avec la version Suntools.

10.1 Les différences entre xjpdw et jpdraw

Il faut d'abord s'assurer que c'est bien la fenêtre xjpdw qui est active. Pour cela il peut être nécessaire de cliquer avec le 3eme bouton dans la fenêtre de xjpdw.

Ensuite une première différence est que les menus apparaissent en utilisant le bouton de gauche. La présentation des menus est différentes. Toutes les instructions apparaissent, séparés par thèmes par des lignes. Le choix courant est en général indiqué par un "x".

La principale différence en dehors de cela vient de la non utilisation du 3eme bouton qui est remplacé par un double click sur le 2eme et par la non utilisation du maintien d'une touche pressée pendant une commande. En général on agit sur la touche, on relache, on agit et on sort du mode spécial en appuyant de nouveau sur la touche.

L'autre différence par rapport à la version Sunview est aussi que l'on n'a plus de menu dans les icônes. Les choix possibles apparaissent en cliquant successivement dans l'icône ou en utilisant les touches R.

Enfin on utilise beaucoup de fenêtre auxiliaire qui apparaissent à l'écran avec une liste de commande. Il suffit de cliquer avec le bouton de gauche de la souris pour valider un choix.

Le reste est indiqué dans la liste suivante:

- Il n'existe pas de transformation des symboles mathématiques en un caractère unique comme dans la version Sunview.
- Polygone: on ferme en faisant un double click sur le bouton du milieu.
- Cercle: les différents modes de création des cercles apparaissent en cliquant dans l'icônes ou en utilisant les touches R1 et suivantes.

- Ovale: les différents modes d'ovales apparaissent en cliquant dans l'icône ou en utilisant les touches R1 et suivantes.
- Marques: les différents marques apparaissent en cliquant dans l'icône ou en utilisant les touches R1 et suivantes.
- Spline, Bspline, Polynome: sont tracés lorsque l'on fait un double click sur le bouton du milieu. Le menu de choix entre les différentes courbes apparaît sur le bouton "C"
- Dans le tracé des rectangles le mode carré ou rectangle est indiqué dans un petit bouton marqué "C" ou "R".
- Autres pointillés: les pointillés n'apparaissent que si l'on a placé la souris à l'intérieur du panel des pointillés.
- Association: la primitive sélectionnée avec le bouton du milieu est associée.
- Grand déplacement: appuyer une fois sur L9, relâchez, cliquer dans la nouvelle position. Pour sortir de ce mode appuyer une nouvelle fois sur L9.
- Style des lignes: en cliquant dans le bouton "—" on fait apparaître un tableau où l'on peut indiquer le style voulu. Sortir par FIN.
- Modification de style: idem qu'au dessus.
- Déplacement fin: en maintenant les touches R appuyé la primitive continue à se déplacer.
- Changement de forme: appuyer une fois sur L10, relâchez, Ajuster la forme. Pour sortir de ce mode appuyer une nouvelle fois sur L10. Pour les textes le nouveau texte est validé en cliquant dans le bouton emacs. Les commandes emacs sont étendus.
- Rotation: les rotations ne se font qu'avec le slider.
- Copie d'écran, Raster: on crée une image X.
- Inrimage: inopérant.
- La couleur est possible à l'aide des sliders rouge, vert, bleu qui apparaissent. Le nombre de couleurs est limité.
- il est impératif que des fonts soit déclarés. En effet si Suntools admet qu'une font ne soit pas définie (il la remplace par la font utilisée par ailleurs), X est beaucoup moins tolérant. Si les fonts standards ne sont pas trouvées JPdraw va tenter de les remplacer au mieux. Mais si aucune font n'est trouvée le programme abandonne. Il faut alors déclarer des fonts existantes dans le configurateur (voir chapitre suivant).

Chapitre 11

Installation de $\mathcal{J}\mathcal{P}\mathcal{d}\mathcal{r}\mathcal{a}\mathcal{w}$ 11.1 Installation de $\mathcal{J}\mathcal{P}\mathcal{d}\mathcal{r}\mathcal{a}\mathcal{w}$ *Installation*

$\mathcal{J}\mathcal{P}\mathcal{d}\mathcal{r}\mathcal{a}\mathcal{w}$ utilise un environnement qu'il est nécessaire de définir si vous voulez pouvoir utiliser toutes ses potentialités. Cet environnement sera appelé une *configuration*. La configuration doit décrire le moyen d'accéder à divers fichiers:

Configuration

- `hdcp`, `hdpc`: deux scripts qui permettent de faire des copies d'écran pour les SUN monochromes et les SUN couleurs. Ces scripts font simplement appel à `screeendump` pipé sur une sortie imprimante par `lpr`. Il faut donc ajuster le nom de l'imprimante dans le script.
- `header.latex`: ce fichier contient une en-tête $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ standard. Il sera concaténé avec le fichier $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ créé par $\mathcal{J}\mathcal{P}\mathcal{d}\mathcal{r}\mathcal{a}\mathcal{w}$ pour obtenir le fichier `bidon.ltex`, directement compilable par $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$.
- `header.end.ltex`: ce fichier contient simplement une fin de fichier $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$. Il sera concaténé avec les précédents.
- `french_sentence`, `english_sentence`: les textes qui apparaissent dans $\mathcal{J}\mathcal{P}\mathcal{d}\mathcal{r}\mathcal{a}\mathcal{w}$, français ou anglais. En effet tous les textes apparaissant à l'écran sont définis dans un fichier externe et par simple traduction de ces fichiers il est possible de créer une version utilisant un autre langage que le français.

L'accès à ces fichiers est fondamental pour certaines commandes de $\mathcal{J}\mathcal{P}\mathcal{d}\mathcal{r}\mathcal{a}\mathcal{w}$. Si ces fichiers ne sont pas accessibles le programme émettra au moment du lancement un certain nombre de message qui donneront le nom des commandes inutilisables du fait de la mauvaise installation.

Ensuite on doit pouvoir accéder à des fonts pour l'affichage des textes à l'écran dans le style choisi. Toutefois si les fonts ne sont pas installés le système utilisera la font standard. Les changement de style ne seront pas visibles à l'écran mais seront toutefois corrects dans le fichier $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ généré.

Puis on peut indiquer le nombre maximum de point que compte une fonction de bezier (car cela dépend de votre Installation). On peut aussi indiquer la syntaxe utilisée pour insérer un fichier Postscript dans le fichier `.dvi`. Enfin il est aussi possible d'indiquer un chemin pour accéder à des programmes écrit dans le style du chapitre **Porgrammation**.

11.1.1 Le fichier configurateur

La configuration est définie dans un fichier configurateur. JPdraw utilise la variable d'environnement JPDRAW pour trouver le configurateur. Il est donc nécessaire de définir cette variable d'environnement.

Dans ce fichier on redéfinit les variables de la manière suivante: les commandes commencent toute par un # suivi d'un mot-clé qui indique la variable à changer. Sur la même ligne on indique la nouvelle valeur de la variables. Les mot-clés sont:

PS, BEZIER, LIB, FONT, DVI->PS, BIB

qui permettent respectivement d'indiquer une nouvelle syntaxe pour l'inclusion de Postscript, le nombre de point par centimètre des courbes de Bezier, le chemin d'accès aux fichiers, les fonts, la nature du programme qui permet de transformer les fichiers .dvi en fichier Postscript et enfin le chemin d'accès aux module de programmation. Dans le cas des fonts le mot-clé est suivi par un autre mot-clé indiquant le nom de la font concernée et sa taille: STANDARD, GRAS ITAL, MATHA, MATHB. On peut placer dans le configurateur des commentaires en utilisant la syntaxe du C. Le champ LIB indique la directory où l'on trouve les différents fichiers nécessaires à JPdraw. Le champ SENTENCE indique le nom du fichier de texte. Les champs FONT MATHA, FONT MATHB indique les fonts utilisés pour transcrire les symboles mathématiques. Il est nécessaire d'utiliser ceux fournis avec JPdraw. Noter que pour la version X de JPdraw on utilise le mot-clé SENTENCE.X. Le même fichier configurateur fonctionne donc indifféremment pour la version Sunview et X. Le champ BIB doit indiquer le nom d'une directory contenant des programmes d'extension .fi exécutable (c'est-à-dire non paramétré), voir le chapitre programmation.

11.1.1.1 Inclusion du Postscript

Malheureusement il n'y a pas de manière standard de inclure un fichier Postscript dans un source L^AT_EX car cette manipulation dépend du programme qui permet de convertir le fichier .dvi en fichier postscript. Ainsi si l'on utilise le dvi2ps fourni avec JPdraw la syntaxe est:

```
\special{ps:plotfile [fichier postscript]}
```

Pour permettre une utilisation avec un programme différent on dispose de différentes possibilités défini dans le configurateur. La première consiste à définir dans le configurateur la syntaxe de la phrase qui sera utilisée dans le fichier L^AT_EX. Ceci se fait avec l'instruction PS. Par exemple pour retrouver la syntaxe du dvi2ps on mettrait:

```
# PS \special{ps:plotfile %s}
```

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 utilisera:

```
# PS \epsffile{%s}
```

Cela n'est malheureusement pas suffisant car certains de ces programmes n'admettent que des formes bien données de fichier Postscript. Pour avoir un contrôle minimal sur le Postscript généré on utilise le mot-clé

DVI->PS

avec la syntaxe:

DVI->PS [nom de programme]

où [nom de programme] est soit dvi2ps, soit dvips, soit dvitops. Dans tous les cas sera placé en tête du fichier 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 placé 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 %!

Faisons une petite remarque sur le configurateur. Lors de l'écriture d'un texte important il peut être intéressant de créer une directory ne contenant que les figures. On peut alors placer un configurateur dans cette directory de façon à ce que le chemin indiqué pour trouver le fichier Postscript soit automatiquement mis à jour. Par exemple dans la directory **Text** vous avez créé une directory **Figures**. Dans cette directory vous placez un configurateur contenant l'instruction:

```
# PS \special{ps:plotfile Figures/%s}
```

Après avoir utilisé la commande `export JPDRAW=configurateur` vous pouvez lancer `JPdraw`. Ainsi vous n'avez pas à mettre à jour le chemin d'accès aux fichiers Postscript.

11.1.1.2 Les fonts

Les fonts sont supposées être définies dans la directory `/usr/lib/fonts/fixwidthfonts`. En standard on utilise les fonts suivantes:

	10pt	12pt	14pt	16pt	18pt	24pt
standard	serif.r.10	serif.r.12	serif.r.14	serif.r.16	cour.r.18	cour.r.24
gras	cour.b.10	cour.b.12	cour.b.14	cour.b.16	cour.b.18	cour.b.24
italique	cour.r.10	cour.r.12	cour.r.14	cour.r.16	cour.r.18	cour.b.24

11.1.1.3 Exemple

La présence des champs suivants est nécessaire: LIB, SENTENCE. Un exemple de fichier configurateur est le suivant:

```
/*-----
inclusion de fichier Postscript
-----*/
# PS \special{ps:plotfile %s}
/*-----
nombre de point par cm des courbes de bezier
```

```

-----*/
# BEZIER 40
/*-----
  la directory librairie
-----*/
# LIB /usr/local/lib/macdraw
/*-----
  les fonts
-----*/
# FONT GRAS16 /usr/lib/fonts/fixedwidthfonts/serif.r.10

# SENTENCE /u/cygnusx1/0/prisme/merlet/Jpdraw/Configurateur/french_sentence
# SENTENCE.X /u/cygnusx1/0/prisme/merlet/Jpdraw/Configurateur/french_sentence.X
# FONT MATHA /u/cygnusx1/0/prisme/merlet/Jpdraw/Configurateur/math.10
# FONT MATHB /u/cygnusx1/0/prisme/merlet/Jpdraw/Configurateur/math1.10
/*-----
Le chemin pour acceder aux programmes
-----*/
# BIB /u/cygnusx1/0/prisme/merlet/Jpdraw/Bib

```

Le nombre de point par centimètre des courbes de Bezier est fixé à 40 alors que la syntaxe pour insérer des fichiers Postscript est donnée par `\special{ps:plotfile %s}`.

11.2 Installation globale

Un makefile est fourni sur la bande JPdraw. Il permet l'installation à la fois de JPdraw et du programme dvi2ps, compatible JPdraw, ce qui n'est généralement pas le cas de tous les dvi2ps. Dans ce makefile mettez à jour les variables suivantes: JPDRAW, JPDRAWBIN, LPLAINEXT. Puis, pour la création du dvi2ps il faut mettre à jour la variable FONTAREA qui indique où se trouve les fonts utilisées par votre imprimante. Si vous utilisez des fonts de type gf au lieu de pxl faites la modification indiquée dans le makefile sur la variable DUSEPXL. Enfin vous avez à traiter la variable LIB_JPDRAW. Deux cas se présentent: si vous avez créé une directory /usr/local/lib/macdraw passez à l'étape suivante. Sinon vous avez une variable d'environnement qui indique où vous voulez mettre les fichiers hdcp, hddpc, header_latex, header_end.ltex. Mettez alors en commentaire la définition de la variable LIB_JPDRAW du makefile. Il suffit maintenant de taper la commande `make install` pour installer correctement JPdraw et créer le programme dvi2ps.

11.3 Sauvegarde de l'état

Conserver
l'état de
JPdraw

Si vous définissez une variable d'environnement `JPDRAW_STATE` comme étant `ON` et que vous quittez normalement JPdraw il y a création d'un fichier (`.jpdraw.state`) décrivant l'état courant de JPdraw, de manière assez complète. Si vous relancez JPdraw 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`.

Chapitre 12

Arguments, Bugs et Résumé

Dans ce chapitre on va indiquer quelles sont les arguments que peut prendre JPdraw au moment du lancement du programme, comment on peut traiter les premières bugs (celle qui sont le plus souvent dues à l'opérateur, pas au programme) et l'on va donner un résumé des touches fonctions.

12.1 Arguments

On peut indiquer plusieurs type d'argument à ce programme.

12.1.1 Un fichier .macdraw

Le 1^{er} 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).

```
jpdraw toto si il existe un fichier toto.macdraw  
mais jpdraw toto.ps.macdraw pour le fichier toto.ps.macdraw
```

12.1.2 Couleurs

On peut indiquer les couleurs de la fenêtre JPdraw selon le codage classique:

- Wb rouge vert bleu : la couleur du fond
- Wf rouge vert bleu : la couleur du 1er plan
- Wg : si les sous fenêtres héritent des couleurs
- WH : pour avoir une liste des arguments possibles. Cette commande permet aussi d'afficher les différents types d'arguments pour JPdraw.

12.1.3 Pour les paraboles

On peut rajouter de plus un autre argument optionnel par "-P xx" où xx indique le nombre de points par cm qui va être utilisé pour la création des primitives paraboliques. Ce nombre est fixé par défaut à 40.

12.1.4 Un fichier fonction

Le 3^{ème} type d'argument que l'on peut donner revient à entrer directement une fonction utilisateur écrite en meta-language. La syntaxe est alors:

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

où x1 représente la largeur en cm de la boîte 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 boîte. Les arguments de -E sont en unité de la fonction.

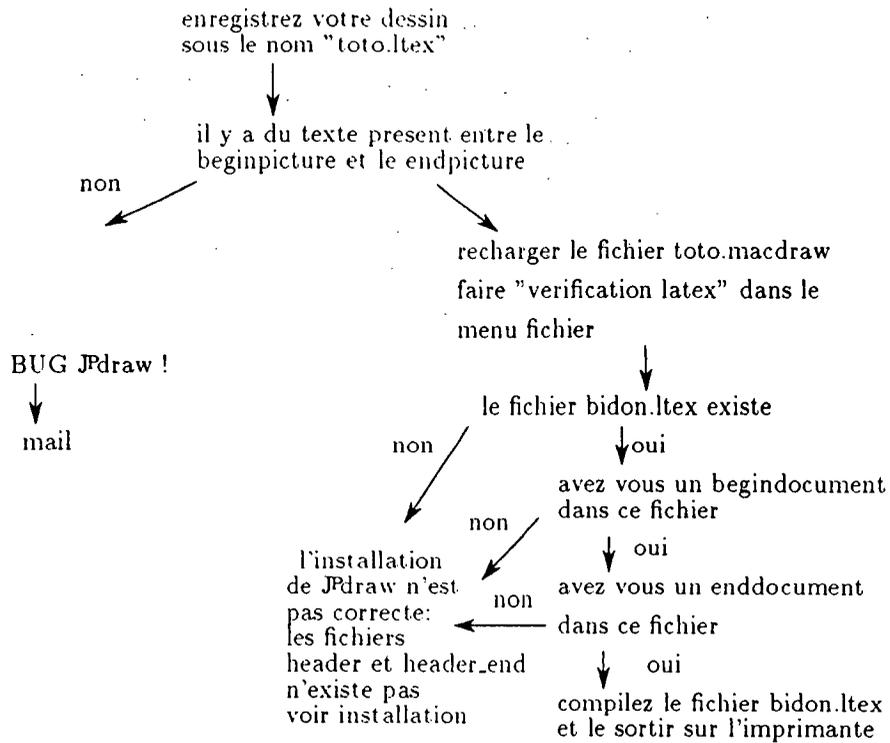
-O permet de fixer la position du coin supérieur gauche de la boîte (par défaut il est placée à 2cm en x et y du coin supérieur gauche du canevas). Les arguments de -O sont en cm.

12.2 Les bugs qui n'en sont pas

Dans l'étape d'apprentissage l'expérience a montré qu'il y avait une première phase critique. Après avoir fait son premier dessin la sortie ne donne rien, pas de dessin ou un espèce de paté informe ou bien L^AT_EX crachent immédiatement des tonnes de message d'injure ou certains ordres ne s'exécutent pas. Pas de panique, la même expérience a montré que la plupart de temps ce démarrage laborieux n'est pas du à une bug du programme mais plutôt à un défaut d'installation ou d'utilisation (il doit cependant rester des bugs bien sûr: il y a une centaine de commandes possibles et je ne suis vraiment pas sûr d'avoir essayé toutes les combinaisons possibles...auquel cas un mail s'impose).

Un cas de figure constaté est que vous avez configuré votre système de manière particulière pour la commande Unix cat auquel fait appel JPdraw. C'est le cas en particulier si vous avez mis à jour la variable "noclobber" de "csh" qui va gêner l'usage de cette commande. La seule commande Unix utilisée par JPdraw est cat. Une autre petite remarque: n'oubliez pas que dans la version avec les extensions vous ne pouvez pas visualiser votre dessin à l'aide d'un previewer du type xdvi. Seul une sortie papier donnera le dessin correcte. Il existe toutefois une version du previewer texsun qui permet de visualiser une version à peu près correcte des dessins JPdraw.

Si il vous arrive que le dessin ne sorte pas suivez la check-list suivante. On va tout d'abord chercher à savoir si c'est le dessin qui est à mettre en cause en l'isolant et en le compilant séparément.



Une dernière remarque. Si vous vous rendez compte que \LaTeX donne une erreur sur une instruction du type \bezier c'est que le nombre de point dans la courbe de bezier est trop important. Vous pouvez alors le diminuer dans l'instruction correspondante. Si cela se reproduit fréquemment vous pouvez diminuer le nombre maximum de point permis (voir page 77).

12.3 Résumé de l'utilisation des touches fonctions et des raccourcis

Pour utiliser une touche fonction 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 tete 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	cercle	supprime l'arc de cercle externe
L8	rectangle	associe les primitives internes du rectangle
L8	ligne	force la pente de la ligne a une pente \LaTeX standard
L8	rotation	fait apparaître un curseur pour indiquer l'angle
L8	cercle	supprime l'arc de cercle interne
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 l'icône de sélection est active 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
m	masquage du presse papier
o	si icône de texture active valide la texture
p	passer la sélectionnée au premier plan
q	affiche ou masque le quadrillage
r	affiche ou masque les règles en centimètres
s	passer la sélectionnée au second plan
[0-9]P	exécution du programme [0-9]

Chapitre 13

Conclusion

Vous allez sans doute rencontrer quelques problèmes au cours de l'utilisation de JPdraw. 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

jpdraw 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)
	(4)	1 = texte souligné
	(5)	1 si c'est un commentaire
	(6)	angle de rotation en radian
1	(0,1)	x,y début ligne
	(2,3)	x,y fin ligne
	(4,5)	penne en x penne en y (valeur \LaTeX)
	(6)	0 ou 1 si penne compatible \LaTeX
	(7)	1=vecteur, 2=pointillé 3=doublé vecteur
	(10)	si ligne de texture numero 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 dixième=style, si >100 souligné
	(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)	
(6)	angle de rotation	
(7)	1 si cercle avec 2 lignes	
(10)	0 =trait plein, 1=trait pointillé	

type	para{}	nature
4	(0,1)	x,y centre
	(2,3)	x,y du rectangle englobant
	(4)	l= dashed
	(5)	l pour rester compatible avec Jpdraw _vl 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 meme chose que le positif mais avec flèche
5	(0)	l= côté en pointillé
	(1)	0 sinon 2 si c'est une flèche blanche 3= flèche noire 4 si c'est une parabole
	(2)	len si flèche, l(2) si parabole a axe vertical(horizontal)
	(3)	dep si flèche, si parabole coeff. a de la parabole en unité page
	(4)	wid si flèche, si parabole angle de rotation
	(5)	l= ligne non fermée
	(6)	si flèche pente x ($\text{\textcircled{P}}\text{X}$), si le polygone appartient a un tracé crayon alors numero du tracé si parabole x origine de la parabole (unité écran)
	(7)	si flèche pente y ($\text{\textcircled{P}}\text{Y}$) 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 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 nulle.

dans la structure primit le texte est toujours rempli, si le texte est inutile on place le mot "ceci_est_le_message_pour_les_primitives_sans_texte".

Ces structures sont sauveés dans le fichier .macdraw sous la forme:

```
nombre de primitives, puis pour chaque primitive
type visibilite
para{0..11}
si polygone :
numero polygone
nombre de point polygone
coordonnees point polygone
texte
```

puis ordre de trace (liste des numeros de primitive)

puis dimensions du dessin sur une ligne

14.2 Annexe 2: Les nouvelles primitives du mode \LaTeX étendu

On donne ici les primitives de l'environnement `picture` dans le mode \LaTeX étendu. Les arguments entre crochet sont optionnels

- `setunitlength[unitlength]`, permet d'effectuer un grossissement sur l'ensemble du dessin
- `PSarrowhead[*]($\Delta x, \Delta y$){len}{wid}{dep}`: les têtes de vecteur (si * l'intérieur est en noir).

- **PSoval**{*}{width}{height}: les ellipses, en noir si *.
- **PStilt**(Δx Δy){objet}: tourne objet selon la nouvelle pente.
- **PSpath**(x0,y0){(x1,y1)(x2,y2)...}: connecte les différents points par des lignes.
- **bezier**{nb}(x1,y1)(x3,y3)(x2,y2): Trace une parabole passant par les points 1 et 2 et dont les tangentes en ces points sont 13 et 23. nb est le nombre de points utilisés pour le tracé.
- **PSpoly**(couleur)(x1,y1){(x2,y2)(x3,y3)...(xn,yn)}: Pour tracer un polygone en niveau de gris. couleur est le pourcentage de blanc dans le niveau de gris(1=noir, 100=blanc). Si ce pourcentage est négatif on ne trace pas la bordure.
- **PScercle**(couleur)(dashed){diam}: Pour tracer un cercle en niveau de gris. Si dashed vaut 1 le bord est pointillé.
- **PSellipse**(couleur)(dashed){width}{height}: Pour tracer une ellipse en niveau de gris.
- **PSrectangle**(couleur)(dashed)(angle){width}{height}: Pour tracer un rectangle en niveau de gris. angle est l'angle de rotation du rectangle en degré.
- **PSarc**(couleur)(dashed){diam}{angle1}{angle2}: Pour tracer un arc de cercle en niveau de gris. angle1 et angle2 en degré définissent l'arc.

14.3 Appendice 3: les caractères spéciaux traités par JPdraw

JPdraw dispose de deux fonts qui permettent de transcrire certains symboles spéciaux de \LaTeX en symbole équivalent à l'écran. Les tables suivantes donnent les correspondances entre la valeur hexadécimale du caractère et le symbole correspondant, successivement pour les fonts math.10 et math1.10.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0						‡	Π	ä	λ	◀	⊂	⊆	∃			
1	≡	ℝ	×	ℝ	ℝ	≡	α	≡	≡	×))	←	→	→	↔
2	↑	↓		λ	≤		γ]	√	∧	≠					
3	∅											[x		
4			⊥	⊂	⊆	ε		∇	Δ	∇	◊		⊥		⊥	
5	*	≡			~	↓	⊂	ε	⊂	⊥	⊥	→	∇			
6	◊	é	è	è	è	î	î	û	±	±	*		•	⊆	⊂	⊂
7	◁	▷	△	▽	⊕	⊖	⊗	⊙	⊙	⊙	†	{	⊥	}	\$	⊥
8	—	—	~	↑												

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0					↑	↓	↓	↓	/	\	/	\	ℝ	h	i	j
1	◀	ρ	ℝ	Ω	U	#	%	&	...	:	·	à	ç	§	¶	©
2	\$	β	γ	δ	ε	ε	ζ	η	θ	∂	ι	κ	λ	μ	ν	ξ
3	π	ω	θ	σ	ς	τ	φ	φ	χ	ψ	Γ	Δ	Λ	Ξ	Π	Σ
4	Υ	Φ	Ψ	Ω	√	∠	∇	∇	∇	∇	∇	∇	∇	∞	□	♣
5	♠	♠	♠	♠												♥

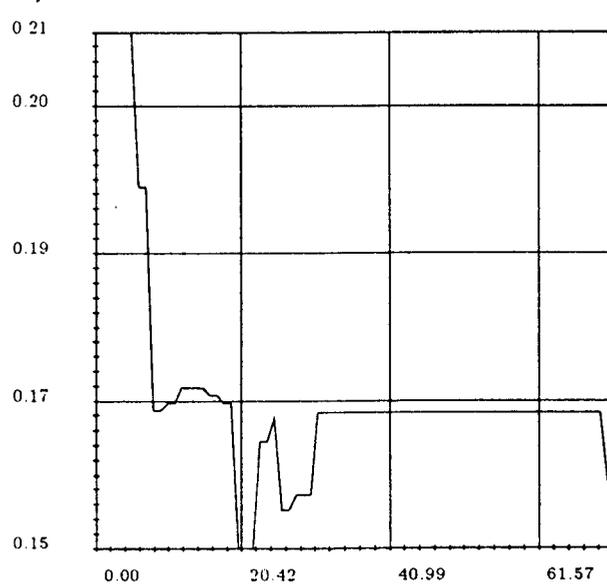
14.4 Quelques exemples de dessin

14.4.1 Tracé de fonction

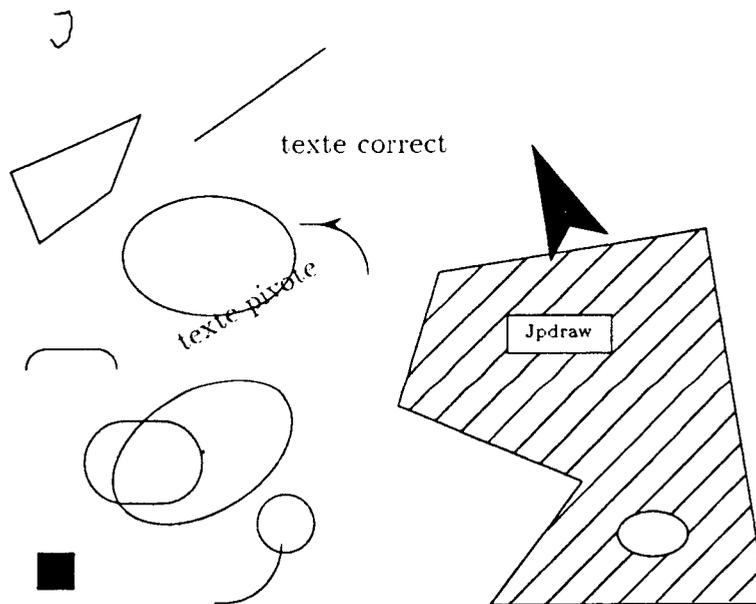
On dispose de 635 couples de coordonnées de points résultant d'une expérience en robotique dans le fichier y.7. Le méta-langage permet de créer les segments joignant ces points. La commande a été lancée par:

```
jpdraw -X 9 -Y 5 -F y.7
```

qui a créé automatiquement le cadre et le tracé de fonction. Le quadrillage ainsi que les valeurs des abscisses et des ordonnées ont été placés automatiquement ultérieurement.

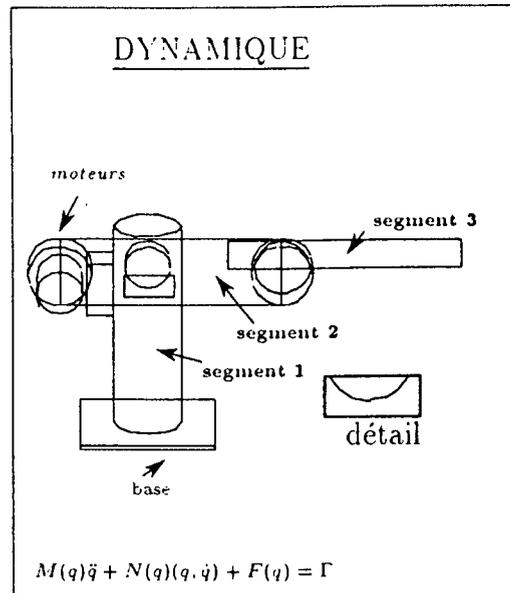


14.4.2 Quelques primitives



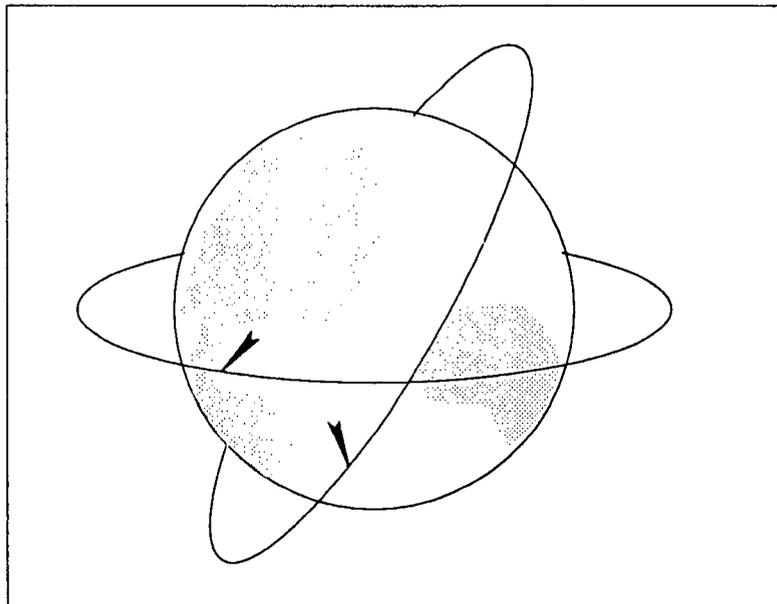
14.4.3 Encore plus...

Comment on peut interfacer un système de CAO (écrit par B. Faverjon) et JPdraw. On a obtenu les détails en associant les primitives dans le rectangle en gras, en les copiant puis en zoomant sur les associées. Les textes ont été rajoutés à la main.



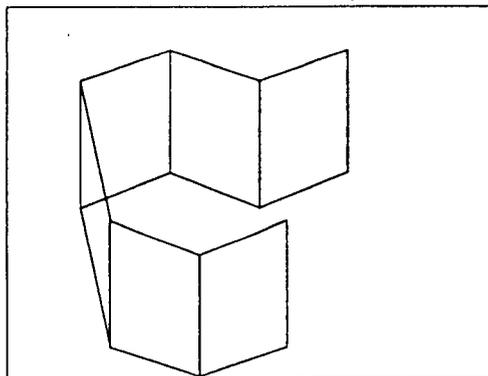
14.4.4 Rotations, satellite et masque

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 plaçant un polygone dessus, que l'on a texturé en blanc sans bordure. On aurait aussi pu utiliser les ciseaux.



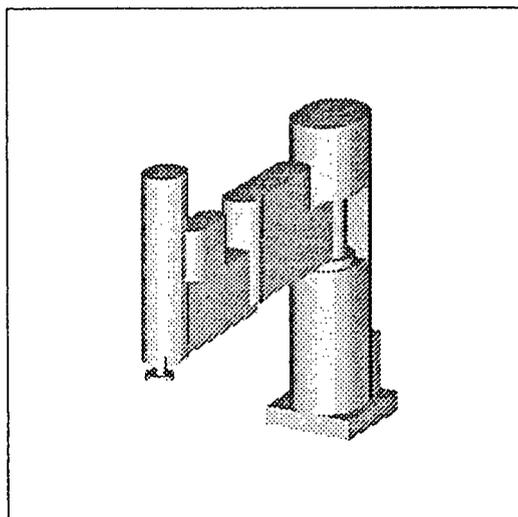
14.4.5 Jeu de miroir

Les murs de cette curieuse pièce ont été créés à partir du même élément de base répété successivement dans des miroirs.



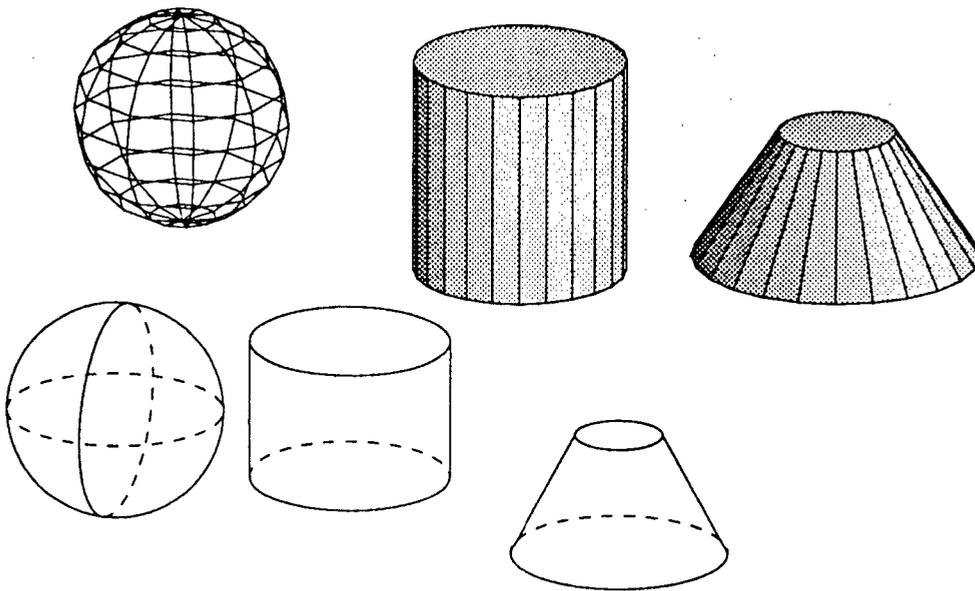
14.4.6 Interface avec un programme de CAO

Le dessin du robot ci-dessous est le résultat de la lecture d'un fichier utilisateur créé par un système de CAO.



14.4.7 Module 3D

Voici quelque dessin que l'on peut obtenir directement en utilisant le module 3D.



14.5 Annexe 3: Comment faire pour....

14.5.1 Affichage

- comment conserver un état de JDraw : page 60
- comment obtenir le centre d'une primitive: page 24
- comment redessiner tout: page 7
- comment voir sa page en taille réelle: page 42

14.5.2 Création

- le pinceau: page 18
- comment créer un arc d'ellipse: page 48
- comment créer un arc de cercle: page 17
- comment créer un cercle: page 17
- comment créer un ovale ou une partie d'ovale: page 18
- comment créer un polygone ou une ligne polygonale: page 16
- comment créer un polygone régulier: page 22
- comment créer un rectangle: page 7
- comment créer une courbe polynomiale: page 19
- comment créer une courbe spline: page 19
- comment créer une courbe B-spline: page 19
- comment créer une courbe tracée à la souris: page 19
- comment créer une ellipse: page 8
- comment créer une marque noir dans le dessin: page 18
- comment créer et utiliser ses propres marques: page 19
- comment créer une parabole: page 20
- comment créer une primitive de texte: page 15
- comment créer une tête de vecteur: page 22

- comment définir le type de primitives que l'on va tracer: page 7
- comment diminuer le nombre de point utilisé pour définir une parabole: page 20
- comment fixer la taille des caractères d'une primitive de texte: page 41
- comment forcer le tracé d'un carré: page 21
- comment modifier l'épaisseur des traits des primitives: page 23
- comment modifier le style des traits des primitives: page 22
- comment modifier l'épaisseur des traits épais existants: page 23
- comment supprimer un point d'un polygone en cours de création: page 17
- comment supprimer un polygone en cours de création: page 17
- comment tracer des polygones pointillés en mode fonction: page 56
- comment tracer une ligne ou un vecteur: page 7
- comment forcer une ligne à avoir une pente compatible avec \LaTeX standard: page 21
- correspondance texte du dessin, texte \LaTeX : page 16
- le style des textes: page 16
- les différentes manières de créer un cercle: page 17
- nombre maximum de sommet d'un polygone: 30: page 17

14.5.3 Divers

- Résumé de l'utilisation des touches fonctions: page 84
- boutons et menus: page 3
- comment afficher des règles graduées en centimètre: page 42
- comment aligner des primitives sur la réglure: page 44
- comment aligner des primitives: page 41
- comment annuler le dernier ordre : page 45
- comment connaître la taille totale de son dessin: page 42
- comment connaître le nombre total de primitives: page 6
- comment faire apparaître un quadrillage dans le canevas: page 43
- comment faire une copie d'écran: page 37
- comment installer JPdraw : page 77
- fenêtre de choix: page 4
- la Réglure: page 44
- la commande Aligner la Selectionnee sur la Reglure: page 44
- la commande Aligner les Associees sur la Reglure: page 44
- la commande Aligner Tout sur la Reglure: page 44
- la commande Modifier la Reglure: page 44
- la commande Modifier la Reglure(cm): page 44
- lancer le programme: page 5
- le mini-buffer: page 6
- les arguments de JPdraw : page 81
- les bugs: page 82
- les fenêtres de JPdraw : page 4
- les fenêtres texte: page 4
- les icônes: page 4
- usage des boutons: page 3
- usage des scrollbars: page 6

14.5.4 Fichiers

- charger un fichier au lancement du programme: page 10
- comment charger un fichier dans Jpdraw : page 10
- comment créer un fichier Postscript contenant le dessin: page 34
- comment créer un fichier contenant le dessin et compilable par \LaTeX : page 12
- comment faire imprimer un fichier trop gros pour \LaTeX : page 32
- comment générer un fichier compatible avec le \LaTeX standard: page 35
- comment rajouter un dessin déjà existant au dessin: page 10
- comment sauvegarder le dessin dans le fichier courant: page 9
- comment sauvegarder le dessin dans les fichiers: page 8
- comment sauvegarder le dessin sous un nom donné: page 9
- comment sauvegarder son dessin sous forme d'image: page 36
- comment sauvegarder le dessin en postscript, texte en \LaTeX , sous un nom donné: page 37
- comment supprimer les primitives détruites dans un fichier: page 39
- date de création du fichier courant: page 9
- l'album: page 10
- la commande Sauver le Dessin en Postscript et le Texte en LATEX: page 34
- la commande Sauver le Dessin en Postscript et le Texte en Latex: page 34
- le format .macdraw: page 87
- les fichiers .macdraw: page 8
- les fichiers générés par Jpdraw : page 8
- nom du fichier courant: page 9
- nom du fichier créé par défaut: page 8
- que se passe-t-il lorsque l'on écrase le fichier courant: page 9
- sauvegarde automatique: page 10
- comment changer le temps entre deux sauvegardes automatiques :page 43
- comment modifier la fréquence des sauvegardes automatiques: page 43

14.5.5 Fonction

- charger un fichier utilisateur: page 52
- comment définir un dessin dans un fichier: les fonctions utilisateurs: page 51
- comment fixer l'échelle d'un dessin: page 60
- comment introduire des couleurs dans un fichier fonction : page 60
- comment modifier le quadrillage standard: page 59
- comment tracer un quadrillage après avoir défini une fonction: page 59
- exemple de fichier fonction: page 58
- le mode arc: page 54
- le mode B-spline: page 57
- le mode cercle: page 54
- le mode connecte: page 53
- le mode deconnecte: page 53
- le mode polygone: page 56
- le mode rectangle: page 56
- le mode spline: page 57
- le mode texte: page 55
- le mode epaisseur_trait: page 58
- les couleurs: page 60
- les erreurs dans les fichiers fonction: page 58
- les instructions du meta-language: page 52
- les niveaux de gris: page 60

14.5.6 Impression

- Les nouvelles primitives du mode $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ étendu: page 89
- comment définir les dimensions du dessins: page 12
- comment imprimer: page 10
- inclure le dessin dans un fichier $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$: page 11
- les extensions $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$: page 11
- les possibilités graphiques standard de $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$: page 10
- validez les extensions: page 11
- DVIPS, DVI2PS: page 79

14.5.7 Manipulation de primitive

- comment éditer une primitive de texte: page 29
- comment changer la taille d'une primitive: page 27
- comment changer la taille des ellipses, ovales: page 27
- comment changer la taille des polygones: page 28
- comment changer la taille des rectangles, cercles, lignes: page 27
- comment copier une primitive: page 30
- comment copier une primitive avec une homothétie: page 30
- comment détruire une primitive: page 29
- comment détruire une primitive cachée: page 38
- comment détruire toutes les primitives sauf la primitive sélectionnée: page 29
- comment détruire toutes les primitives sauf les associées: page 29
- comment faire tourner un ensemble de primitives: page 29
- comment faire tourner une primitive: page 28
- comment faire tourner une primitive par un slider: page 28
- comment faire un zoom de son dessin: page 43
- comment modifier la taille et le style d'un texte d'un texte existant: page 41
- comment modifier le style des traits des primitives déjà existantes: page 23
- comment pivoter une la commande: page 44
- comment récupérer une primitive détruite: page 29
- comment recommencer un dessin: page 29
- comment translater une primitive (déplacement fin): page 26
- comment translater une primitive (grand déplacement): page 26
- comment translater une primitive (petit déplacement): page 26
- la commande Coller: page 29
- la commande Copier: page 30
- la commande Detruire: page 29
- la commande Detruire Tout sauf la Selectionnee: page 29
- la commande Detruire Tout sauf les Associees: page 29
- la commande Miroir: page 44
- la commande Pivoter a Gauche: page 44
- la commande Zoom: page 43
- la fenêtre emacs: page 30
- la manipulation de primitives: page 26
- le presse-papier: page 29
- les commandes de la fenêtre emacs: page 30
- les miroirs, pivot et le zoom: page 31
- masquer une primitive à l'aide d'une texture: page 48
- où va la primitive détruite: page 29

14.5.8 Menus

- la commande Regles: page 42
- la commande ANNULER DERNIER ORDRE: page 45
- la commande BIB: page 45
- la commande Verification Latex: page 37
- la commande Verification Latex avec Postscript: page 37
- la commande Vider le Presse-Papier: page 39
- le menu CARACT.: page 41
- le menu DISPOSITION: page 44
- le menu EDITION: page 38
- le menu FICHER: page 36
- le menu PAGE: page 42
- le menu STYLE: page 41
- les commandes Taille Reelle ,Taille Ecran: page 42
- les menus: page 35

14.5.9 Postscript

- Jpdraw et Postscript: page 32
- comment créer un fichier Postscript contenant le dessin: page 34
- comment insérer un fichier Postscript dans du L^AT_EX : page 33
- comment obtenir un fichier L^AT_EX contenant le dessin avec du postscript: page 37
- comment ajuster la taille du dessin Postscript à la taille du dessin L^AT_EXsi la valeur de la variable \unitlength ne vaut pas 1cm: page 40

14.5.10 Sélection et association

- comment associer des primitives: page 25
- comment dissocier les primitives: page 25
- comment dissocier une primitive: page 25
- comment sélectionner une primitive: page 24
- comment sélectionner une primitive recouvrante: page 24
- sélection de primitives du même type: page 40
- l'association de primitives dans un rectangle: page 25
- l'association de primitives du même type: page 25
- l'association de primitives par désignation: page 25
- la notion d'association: page 24
- la notion de sélection: page 24

14.5.11 Texture

- comment définir une texture à niveau de gris: page 48
- comment supprimer une texture: page 48
- comment texturer une primitive : page 46
- créer sa propre texture de lignes: page 47
- les textures: page 46
- les textures à base de lignes: page 46
- les textures à niveau de gris: page 48
- les trous dans une texture de lignes: page 49
- masquer une primitive à l'aide d'une texture: page 48
- comment rendre une texture permanente: page 50

Table des matières

1	Le premier dessin	2
1.1	Introduction	2
1.2	Pour commencer: lignes, rectangles, ellipses	3
1.3	Boutons, Menus et Icônes	3
1.3.1	Boutons et menus	3
1.3.2	Icônes	4
1.3.3	Remarques	4
1.3.4	Les fenêtres de JPdraw	4
1.4	Lancer le programme	5
1.4.1	Tableau de bord	5
1.5	Le tracé de primitives	7
1.6	Ligne et style de trait	7
1.7	Rectangle	7
1.8	Ellipse	8
1.9	Comment sauver dans un fichier	8
1.9.1	Les fichiers générés	8
1.9.2	Dans le fichier par défaut	8
1.9.3	Sauver dans un fichier donné	9
1.9.4	Ecrire dans un fichier existant	9
1.10	Sauvegarde automatique	10
1.11	Charger un fichier	10
1.11.0.1	A l'intérieur de JPdraw	10
1.11.0.2	Au lancement du programme	10
1.11.0.3	Album	10
1.12	Imprimer	10
1.12.1	Les possibilités graphiques de L ^A T _E X	10
1.12.2	Les extensions	11
1.12.3	Installation et utilisation des extensions	11
1.12.4	Incorporez votre dessin	11
1.12.4.1	Insérer un dessin par un éditeur de texte	11
1.12.4.2	Insérer un dessin par <code>\input</code>	12
1.12.5	Les dimensions	12

1.12.6	La commande Vérification Latex	12
1.12.7	Imprimer	12
1.13	Utilitaires	12
1.13.1	Le module 3D	13
1.13.1.1	Ajustement de la vue 3D	13
1.13.1.2	Sphère	13
1.13.1.3	Cube	13
1.13.1.4	Cylindre	13
1.13.1.5	Tronc de cône	14
1.14	Quitter	14
2	Des primitives plus complexes	15
2.1	Texte	15
2.2	Taille des textes en sortie	16
2.3	Polygone	16
2.4	Cercle	17
2.5	Arc de cercle	17
2.6	Ovale	18
2.7	Marques	18
2.7.1	Les marques standards	18
2.7.2	Définir et utiliser une marque	19
2.8	Le crayon	19
2.8.1	Dessin à la souris (Souris)	19
2.8.2	Courbe interpolée à partir de points (Spline)	19
2.8.3	B-splines	19
2.8.4	Approximation polynomiale (Polynome)	19
2.8.5	Parabole (Parabole)	20
2.9	Usage avancé des créations de primitives	21
2.9.1	Ligne	21
2.9.2	Rectangle	21
2.9.2.1	Tracé d'un carré	21
2.9.3	Polygones particuliers	21
2.9.3.1	Tête de vecteur	22
2.9.3.2	Polygone régulier	22
2.10	Le style des traits des primitives	22
2.11	Modifier un style de trait	23
3	Sélection et association	24
3.1	Icône flèche en haut	24
3.1.1	Sélectionner une primitive	24
3.1.2	Sélection pour primitives recouvrantes	24
3.2	Association	25
3.2.1	Associer plusieurs primitives	25

4 Manipulation de primitives	26
4.1 Translater une primitive	26
4.2 Modifier la taille d'une primitive	27
4.2.1 Cas des rectangles, cercles, lignes	27
4.2.2 Cas des ellipses, ovales	27
4.2.3 Cas des textes	27
4.2.4 Cas des polygones	28
4.3 Rotation	28
4.4 Détruire une primitive	29
4.4.1 Détruire une primitive ou un ensemble de primitives	29
4.4.2 Où va la primitive détruite?	29
4.4.3 Redémarrer	29
4.4.4 Détruire tout sauf les associées	29
4.4.5 Détruire tout sauf la sélectionnée	29
4.5 Correction de texte	29
4.6 Copier une primitive	30
4.7 Copier une primitive avec homothétie	30
4.8 Les miroirs, pivot et le zoom	31
4.9 Couper une primitive	31
4.10 Loupe	31
5 JPdraw et Postscript	32
5.1 Mélanger JPdraw et Postscript	32
5.2 La commande <code>special</code>	33
5.3 Créer les fichiers L ^A T _E X et Postscript	34
5.3.1 La commande <code>Sauver le Dessin en Postscript et le Texte en Latex</code>	34
5.3.2 La commande <code>Sauver le Dessin en Postscript et le Texte en Latex Sous</code>	34
5.4 Textures et textes	34
6 Description détaillée des menus	35
6.1 Le bouton <code>EXTENSIONS</code>	35
6.2 Le Menu <code>FICHER</code>	36
6.2.1 Redemarrer	36
6.2.2 Charger	36
6.2.3 Sauver	36
6.2.3.1 Sauver	36
6.2.3.2 Sauver Sous	36
6.2.3.3 Sauver sous format raster	36
6.2.3.4 Sauver sous format inrimage	36
6.2.3.5 Sauver le Dessin en Postscript et le Texte en Latex	37
6.2.3.6 Sauver le Dessin en Postscript et le Texte en Latex Sous	37
6.2.4 Impression Ecran	37
6.2.5 Verifier	37

6.2.6	Album	37
6.3	Le Menu EDITION	38
6.3.1	Detruire	38
6.3.1.1	Detruire la Primitive Selectionnee	38
6.3.1.2	Detruire Tout sauf les Associees	38
6.3.1.3	Detruire Tout sauf la Selectionnee	38
6.3.2	Eliminer les Primitives cachees	38
6.3.3	Copier	38
6.3.3.1	Copier la primitive sélectionnée	38
6.3.3.2	Copier avec Homothetie	38
6.3.4	Le presse-papier	38
6.3.4.1	Coller	38
6.3.4.2	Afficher le Presse-Papier	39
6.3.4.3	Masquer le Presse-Papier	39
6.3.4.4	Vider le Presse-Papier	39
6.3.4.5	Permuter le presse-papier et les primitives visibles	39
6.3.5	Associer-Dissocier	39
6.3.5.1	Associer Tout	39
6.3.5.2	Associer les primitives du même type	39
6.3.5.3	Dissocier Tout	39
6.3.5.4	Dissocier la Primitive Selectionnee	39
6.3.6	Selection Selective	40
6.3.7	Changer la valeur de Unitlength	40
6.3.8	Formule	40
6.3.9	Aligner	41
6.3.10	Remettre ou enlever Bordure	41
6.4	Le Menu STYLE	41
6.5	Le Menu CARACT.	41
6.5.1	10pt...24pt	41
6.5.2	Texte existant	41
6.5.2.1	Pour tous	41
6.5.2.2	Pour les Associees	42
6.5.2.3	Pour la Primitive Selectionnee	42
6.5.2.4	Pour la Derniere Primitive Entree	42
6.6	Le Menu PAGE	42
6.6.1	Redessiner Tout	42
6.6.2	Regles	42
6.6.3	Taille	42
6.6.3.1	Taille du Dessin	42
6.6.3.2	Taille Reelle, Ecran	42
6.6.4	Quadrillage	43
6.6.5	Zoom	43
6.6.6	Configuration	43
6.6.7	Centrer	43

6.7	Le Menu DISPOSITION	44
6.7.1	Plan	44
6.7.2	Pivot	44
6.7.3	Miroir	44
6.7.4	Réglure	44
6.7.4.1	Aligner la Selectionnee sur la Reglure	44
6.7.4.2	Aligner les Associees sur la Reglure	44
6.7.4.3	Aligner Tout sur la Reglure	44
6.7.4.4	Modifier la reglure	45
6.7.4.5	Ordre Inverse	45
6.8	ANNULER DERNIER ORDRE	45
6.9	Le Menu BIB	45
7	La texture des primitives	46
7.1	Texture	46
7.1.1	Comment texturer une primitive	46
7.1.2	Textures faites de lignes	46
7.1.2.1	Type de texture	47
7.1.2.2	Créer votre texture	47
7.1.3	Texture à niveau de gris et en couleur	48
7.1.3.1	Définir une texture	48
7.1.3.2	Les masques	48
7.1.4	La gomme	48
7.1.5	Les trous dans une texture	49
7.1.6	Remarques sur les textures	49
7.1.6.1	Restriction sur les textures des tracés crayon	50
7.1.7	Rendre une texture permanente	50
8	Les fonctions utilisateurs	51
8.1	Fonction	51
8.1.1	Généralités	51
8.1.2	Les fonctions utilisateurs	51
8.1.3	Comment charger un fichier utilisateur	52
8.1.4	Les fichiers utilisateurs: le meta-language	52
8.1.4.1	Le mode comment	52
8.1.4.2	Le mode connecte	53
8.1.4.3	Le mode deconnecte	53
8.1.4.4	Le mode cercle	54
8.1.4.5	Le mode arc	54
8.1.4.6	Le mode texte	55
8.1.4.7	le mode rectangle	56
8.1.4.8	Le mode polygone	56
8.1.4.9	Mode bspline	57
8.1.4.10	Mode spline	57

8.1.4.11	Le mode epaisseur_trait	58
8.1.4.12	Le mélange des modes	58
8.1.5	Erreur dans les fonctions utilisateurs	58
8.1.6	Tracé d'abscisses et d'ordonnées	59
8.1.7	Echelle fixe	60
8.1.8	La couleur et les niveaux de gris	60
8.1.8.1	Couleur	60
8.1.8.2	Niveau de gris	60
8.2	Résumé des modes	62
9	Programmation	63
9.1	Introduction	63
9.2	Variables et expressions	63
9.2.1	Expressions élémentaires	63
9.2.2	Variables	64
9.2.3	Valeur	64
9.2.4	Coordonnées	64
9.3	Chaîne de caractères	65
9.4	Impression	65
9.5	Style des traits	65
9.6	Entrée de points à la souris	65
9.7	Emission de message vers la console	66
9.8	Création de primitives	66
9.8.1	Texte	66
9.8.2	Cercles, Ellipses	66
9.8.3	Rectangle	66
9.8.4	Ligne	67
9.8.5	Courbes	67
9.9	Manipulation des primitives	67
9.9.1	Destruction de primitives	67
9.9.2	Association, dissociation	67
9.9.3	Translation, rotation	68
9.9.4	Copier une primitive	68
9.9.5	Miroirs	68
9.9.6	Centrer	68
9.10	Programme paramétré	68
9.10.1	Déclaration des arguments	69
9.10.2	Récupération des arguments	69
9.10.3	Unitlength	69
9.10.4	Commentaires	69
9.10.5	Sauver et charger un fichier	69
9.10.6	Fin de programme	70
9.11	Lire une fonction utilisateur	70
9.12	Divers	70

9.12.1	Distance entre deux points	70
9.12.2	Centre d'une primitive	70
9.12.3	Intersection de deux primitives	70
9.12.4	Coordonnées de départ et de fin d'une primitive	70
9.13	Exemples de programme	70
9.13.1	Exemple 1	70
9.13.2	Exemple 2	71
10	X et JPdraw	75
10.1	Les différences entre xjpdw et jpdw	75
11	Installation de JPdraw	77
11.1	Installation de JPdraw	77
11.1.1	Le fichier configurateur	78
11.1.1.1	Inclusion du Postscript	78
11.1.1.2	Les fonts	79
11.1.1.3	Exemple	79
11.2	Installation globale	80
11.3	Sauvegarde de l'état	80
12	Arguments, Bugs et Résumé	81
12.1	Arguments	81
12.1.1	Un fichier .macdraw	81
12.1.2	Couleurs	81
12.1.3	Pour les paraboles	82
12.1.4	Un fichier fonction	82
12.2	Les bugs qui n'en sont pas	82
12.3	Résumé de l'utilisation des touches fonctions et des raccourcis	84
13	Conclusion	86
14	Annexes	87
14.1	Annexe 1: Le format .macdraw	87
14.2	Annexe 2: Les nouvelles primitives du mode L ^A T _E X étendu	89
14.3	Appendice 3: les caractères spéciaux traités par JPdraw	91
14.4	Quelques exemples de dessin	91
14.4.1	Tracé de fonction	91
14.4.2	Quelques primitives	92
14.4.3	Encore plus...	92
14.4.4	Rotations, satellite et masque	93
14.4.5	Jeu de miroir	94
14.4.6	Interface avec un programme de CAO	94
14.4.7	Module 3D	94
14.5	Annexe 3: Comment faire pour....	95
14.5.1	Affichage	95

14.5.2	Création	95
14.5.3	Divers	96
14.5.4	Fichiers	97
14.5.5	Fonction	97
14.5.6	Impression	98
14.5.7	Manipulation de primitive	98
14.5.8	Menus	99
14.5.9	Postscript	99
14.5.10	Sélection et association	99
14.5.11	Texture	99

Index

- 3D ,13
- # ,52,66
- % ,64
- & ,68
- accélérateur ,9
- Afficher le Presse-Papier ,29,39
- afficher les règles ,42
- afficher
 - le presse-papier ,39
- ajouter un dessin ,37
- Album ,10,37
- aligner ,41
 - sur la réglure ,44
- angle de rotation ,28
- ANNULER DERNIER ORDRE ,45
- arc d'ellipse ,48
- arcs de cercle ,17
 - texture ,46
 - vecteur ,17
- arguments ,10,69,80
 - couleurs ,81
 - déclaration ,69
 - fonction ,82
 - liste ,81
 - .macdraw ,81
 - parabole ,82
 - recupération ,69
- arrondis d'ovale ,18
- Association Multiple ,25
- association ,24,25,39,68
 - multiple ,25
 - primitives désignées ,25
 - rectangle ,25
 - selon type ,25
- Associer Tout ,39
- \setunitlength ,40
- \special ,33
- backspace ,15
- barycentre ,28,31
- bezier ,20
- BEZIER ,78
- BIB ,45,63,78
- bidon.ltex ,12,37,77
- BORDURE NOIRE ,48
- bordure ,41
- BoundingBox ,33,79
- boutons ,3
- B-spline ,19
- bugs ,82
- canevas ,5
- carac ,15
- caractères ,65
 - gras ,41
 - italiques ,41
 - spéciaux ,16
 - style ,41
 - taille ,15,41
- carré ,21
- cat ,82
- center ,70
- centre d'une primitive ,70
- Centrer ,43,68
- cercles ,10,17,18,25,27,31
 - arc de ,17
 - en ellipses ,17
 - modes de création ,17
 - texture ,46
- chaîne de caractères ,65
- changement de taille ,27
- changement d'échelle ,60
- Charger ,10,36
 - un fichier ,69
- ciseaux ,12,31
- clignotement ,24
- Coller ,29,38
- commentaires ,15,69

- dans les fonctions ,52
- cône ,14
- configurateur ,78
 - exemple ,79
- coordonnée ,64
- coordonnées de départ et de fin d'une primitive ,70
- copie d'écran ,37
- copier avec homothétie ,30
- copier ,30,68
- correction de textes ,29
- couleur ,48,60
- couper ,29
 - une primitive ,31
- courbes
 - polynomiale ,19
- courbes ,19,67
- crayon ,19
- create ,37
- créer une texture ,47
- csh ,82
- CTRL ,30
- cube ,13
- curseurs ,28,30,37,43,48
- cylindre ,13
- date ,9
- déclaration des arguments ,69
- delete ,15
- dessin
 - à la main ,19
 - ajouter ,10
 - inclure un ,11
 - vérification ,37
- destruction de primitives ,67
- Detruire tout sauf la Selectionnee ,29
- Detruire tout sauf les Associees ,29
- Detruire ,29,38
- détruire
 - le presse-papier ,29
 - tout sauf la sélectionnée ,29
 - tout sauf les associées ,29
 - une primitive ,29
- dimensions ,12,31
 - des caractères ,15
 - des primitives ,42
 - du dessin ,42
 - fichier L^AT_EX,12
 - modifier ,27
- DISPOSITION ,44
- dissociation ,68
- Dissocier la Primitive Selectionnee ,39
- Dissocier la Selectionnee ,25
- Dissocier Tout ,25,39
- dissocier tout ,25
- dissocier une primitive ,25
- dissocier ,25
- distance
 - entre deux points ,70
- dvi2ps ,79,80
- DVI- ψ PS ,78
- dvips ,79
- dvitops ,79
- E ,82
- échelle fixe ,60
- éditer un ancien texte ,30
- éditer un texte ,29
- EDITION ,38
- Eliminer les Primitives cachées ,38
- ellipses ,17,25,27,8
 - texture ,46
- emacs ,29,5
- enlever une bordure ,41
- épaisseur des traits ,7
- epsfile ,79
- Etat ,80
- exécution des programmes ,63
- exemple
 - de configurateur ,79
 - de dessins ,91
 - de fichier fonction ,58,60
 - de fichier Postscript ,33
 - de programmation ,71
- extension .ps ,34
- extensions ,11,35
 - installation ,11
 - utilisation ,11
- F ,63,82
- facette ,13
- Facettes Ombrees ,13
- Facettes ,13
- fenêtre de texte ,4
- fenêtres actives ,4
- FI ,63
- .fi ,63,78