

Visualisation de maillages 2D et 3D et de surfaces 3D sous X

François JOUVE¹

1 Introduction

`xd3d` est un outil graphique capable de visualiser des maillages bi et tridimensionnels, des champs scalaires ou vectoriels associés à ces maillages (sous forme d'isovaleurs ou de flèches), ainsi que des surfaces 3D décrites dans des fichiers ou par leurs équations algébriques. Son utilisation est censée être intuitive. Si elle ne l'est pas suffisamment, il faut s'en plaindre à l'auteur.

2 Logiciels annexes à installer éventuellement

Pour pouvoir utiliser toutes les possibilités de `xd3d`, il vous faut posséder, installés dans un directory contenu dans votre variable `PATH`, un certain nombre de logiciels, qui appartiennent tous au domaine public. Toutefois, si un ou plusieurs de ces logiciels étaient manquant, seule l'option de `xd3d` qui l'utilise serait inaccessible, le reste du programme fonctionnant parfaitement. Voici la liste des programmes externes appelés par `xd3d` :

- `nopo_avoir` : traduction des fichiers `nopo` issus de Modulef.
- `xgraphic` : logiciel de tracé de courbes. Il est appelé lorsqu'on demande des coupes d'une surface 3D (option “!”).
- `xwdtoppm`, `ppm2tiff`, `ppmtogif`, `ppmtopict`, `ppmtorgb3`, `ppmtoxpm` : servent à la traduction d'un fichier bitmap en différents formats (option “|”).
- `gzip` : outil de compression de `gnu`. Utilisé pour sauver une image bitmap compressée (option “|”).

3 Les données

`xd3d` accepte en entrée deux formats de données pour les maillages et deux formats pour les surfaces tridimensionnelles.

3.1 Les formats de maillages

`xd3d` reconnaît les maillages au format “`nopo`” de Modulef. Pour cela, les fichiers doivent impérativement posséder le suffixe `.nopo`. Pour plus de renseignements sur la structure de donnée `nopo`, se reporter à sa doc Modulef préférée. La traduction s'effectuant par l'intermédiaire d'un

¹Centre de Mathématiques Appliquées (CMAP), Ecole Polytechnique, 91128 Palaiseau Cedex, FRANCE.
tel : 01-69-33-46-35, fax : 01-69-33-30-11
Francois.Jouve@polytechnique.fr, [http ://www.cmap.polytechnique.fr/~jouve](http://www.cmap.polytechnique.fr/~jouve)

programme appelé `nopo_avoir` (cf. 2), celui-ci doit être installé pour pouvoir lire des fichiers “`nopo`”.

D’autre part, le format “natif” de `xd3d` se nomme “avoir”. Les fichiers 3D à ce format peuvent prendre n’importe quel nom (l’extension `.avoir3D` est recommandée), tandis que les fichiers contenant des maillages bidimensionnels doivent posséder le suffixe `.avoir2D`. Toutefois, un maillage bidimensionnel plongé dans une structure tridimensionnelle (l’une des coordonnées étant nulle) sera reconnu sous son camouflage et traité comme il le doit.

Les fichiers “avoir” sont des fichiers binaires. Leur format est décrit dans une annexe à la fin de ce document. Lorsqu’un fichier `.avoir` est lu, le programme recherche la présence d’un fichier `.depl` ayant le même préfixe et qui soit compatible avec le maillage (même nombre de nœuds). Les fichiers `.depl` contiennent un champ de déplacement et ont le même format que les fichiers vectoriels attachés aux nœuds (cf. 3.3). Lorsqu’un fichier `.depl` est lu, le maillage déformé est affiché par défaut. On peut revenir au maillage brut en demandant une exagération nulle des déplacements (option `n`).

3.2 Les surfaces tridimensionnelles

`xd3d` permet de visualiser des surfaces tridimensionnelles dont les points sont contenus dans un fichier de type “`c3d`” ou “`xyz`”, ou bien dont l’expression analytique est entrée au clavier. Ce dernier cas est décrit dans un paragraphe suivant (Utilisation).

3.2.1 Fichiers au format “`c3d`”

Les fichiers de données de ce format doivent impérativement posséder le suffixe `.c3d`. Ils peuvent contenir une ou plusieurs surfaces $z = f(x, y)$ définies sur la même grille de points (x, y) . Cette grille n’est pas nécessairement régulière mais elle est structurée. Le format des fichiers `.c3d` est défini par le petit programme Fortran suivant qui écrit un fichier pour des données supposées être en mémoire centrale :

```

      real*4 x(nx),y(ny),z(nx,ny,nbsurf)
      open(1,file='toto.c3d',form='unformatted',recl=1000000)
c
c nx = nombre de points en x
c ny = nombre de points en y
c
      write(1) nx,ny
      write(1) (x(i),i=1,nx),(y(i),i=1,ny)
c
c nbsurf = nombre de surfaces
c
      do ns=1,nbsurf
        write(1) ((z(i,j,ns),i=1,nx),j=1,ny)
      enddo
      close(1)
      end

```

On peut également tracer des surfaces tridimensionnelles en coordonnées sphériques ($r = f(\theta, \phi)$) ou cylindriques ($z = f(r, \theta)$). Par exemple, pour les coordonnées sphériques, le programme s’écrit :

```

      real*4 theta(nt),phi(np),r(nt,np,nbsurf)
      open(1,file='toto.c3d',form='unformatted',recl=1000000)
c
c iopt = 1 --> coordonnees cylindriques
c iopt = 2 --> coordonnees spheriques
c nx = nombre de points en theta
c ny = nombre de points en phi
c
      iopt = 2
      write(1) 0,iopt
      write(1) nt,np
      write(1) (theta(i),i=1,nt),(phi(i),i=1,np)
c
c nbsurf = nombre de surfaces
c
      do ns=1,nbsurf
        write(1) ((r(i,j,ns),i=1,nt),j=1,np)
      enddo
      close(1)
      end

```

3.2.2 Fichiers au format “xyz”

Les fichiers de données de ce format doivent impérativement posséder le suffixe `.xyz`. Ils ne définissent qu’une seule nappe. Il s’agit de fichiers ASCII contenant un triplet de coordonnées (x, y, z) par ligne. Il est possible d’insérer un nombre arbitraire de lignes de commentaires au début du fichier. Les lignes de commentaires commencent par le caractère `#`. A partir du nuage de points lu, `xd3d` tracera la nappe définie par les coordonnées z et le maillage Delaunay de l’enveloppe convexe des points (x, y) . Lorsqu’il y a une trop grande différence entre la taille de l’objet en x , y et z , `xd3d` demande à l’utilisateur pendant la lecture s’il désire appliquer un facteur multiplicatif aux coordonnées z .

3.3 Les champs scalaires et vectoriels

Les champs servant au tracé des isovaleurs (scalaires) ou des flèches (vectoriels) sont contenus dans des fichiers ASCII de structure très simple : les valeurs peuvent être données aux noeuds ou au centre des cellules. Les premières lignes peuvent contenir des commentaires précédés par un `#` en première colonne. La première ligne de commentaire apparaîtra alors automatiquement comme légende sur le dessin (cf. option `g` pour déplacer et changer la légende).

La première ligne du fichier qui n’est pas un commentaire contient le nombre de données (*i.e.* le nombre de noeuds ou le nombre de cellules), puis les valeurs, à raison d’une par ligne pour les champs scalaires, et N par ligne pour les champs vectoriels en dimension N (la dimension de l’espace contenant le maillage *i.e.* $N = 2$ ou $N = 3$).

Notons un cas particulier de fichier de valeurs d’un format différent : les fichiers contenant un ou plusieurs jeux de forces, comme ceux qui sont utilisés par `genpro`, qui doivent impérativement posséder le suffixe `.for` ou `.force` pour être reconnus, sont lus également afin de pouvoir visualiser par des flèches les forces extérieures utilisées pour un calcul. Ils contiennent un ou

plusieurs blocs successifs composés d'un entier NF_i sur la première ligne désignant le nombre de points concernés par le jeu de forces ponctuelles en question, puis la liste des forces (NF_i lignes composée d'un entier – le numéro du nœud – et des N composantes de la force).

4 Utilisation

4.1 Généralités

Pour lancer `xd3d`, tapez

```
xd3d
```

et répondez aux questions, ou bien directement

```
xd3d toto
```

qui cherchera à tracer, dans l'ordre, un éventuel fichier `toto` au format “avoir3D” puis, s'il n'existe pas, un fichier `toto.avoir3D`, puis un fichier `toto.nopo` (au format “nopo”), un fichier `toto.c3d` et enfin un fichier `toto.xyz`. Si aucun de ces fichiers n'existe, `xd3d` vous demandera un autre nom de fichier en affichant à l'écran, pour vous aider, la liste des fichiers possédant des suffixes qui lui parlent.

Pour tracer une surface 3D dont on connaît l'équation, il faut entrer `c3d` à la place d'un nom de fichier ou taper directement :

```
xd3d c3d
```

Deux options sont alors disponibles : le tracé d'une surface dont l'équation est sous la forme :

$$z = f(x, y)$$

ou bien une surface paramétrée de la forme :

$$(x, y, z) = (x(u, v), y(u, v), z(u, v))$$

Pour les surfaces du premier type, tapez son équation en utilisant les variables x et y et la syntaxe Fortran. Par exemple :

```
sin(x*y)*x**2, pour tracer  $f(x, y) = x^2 \sin(xy)$ .
```

Pour les surfaces du second type, entrez successivement les équations de x , y et z . Attention, les paramètres utilisés doivent s'appeler `x` et `y`. Exemple de dialogue pour dessiner une sphère de rayon 1 :

```
>>> Surface z=f(x,y) ou parametree (0/1) ?
1
>>> Nombre de points en U ?
20
>>> Nombre de points en V ?
20
>>> Entrez Umin
0
>>> Entrez Umax
3.1416
>>> Entrez Vmin
0
>>> Entrez Vmax
6.2832
```

```

>>> Tapez l'expression de X ; (une seule ligne, syntaxe Fortran)
sin(x)*cos(y)
>>> Tapez l'expression de Y ; (une seule ligne, syntaxe Fortran)
sin(x)*sin(y)
>>> Tapez l'expression de Z ; (une seule ligne, syntaxe Fortran)
cos(x)

```

Toutes les commandes utilisables sont accessibles en cliquant sur les boutons de la fenêtre de gauche et par leur raccourci-clavier (toujours un seul caractère, sans différenciation entre les majuscules et les minuscules). Les touches actives s'inscrivent en noir et les touches inactives en grisé.

Quelques exceptions à cette règle :

- La touche **Ne bougeons plus** n'a pas de raccourci-clavier.
- La touche **Num** et le caractère **^** concernent l'affichage de numéros du maillage mais ne donnent pas le même résultat (cf. paragraphe sur **^** pour plus de détails).
- “**a**”, “**c**” et “**,**” qui permettent respectivement le zoom arrière, le zoom avant et l'affichage de valeurs au point indiqué par le curseur, n'ont pas de bouton associé pour des raisons évidentes.
- Le petit drapeau français ou américain qui permet de passer d'une langue à l'autre.

Pour trouver comment faire une action précise, on peut évidemment compulser frénétiquement le présent document s'il correspond à votre version de **xd3d** (taper **xd3d --version** pour la connaître), mais il peut être plus rapide de parcourir la fenêtre de gauche avec votre curseur en regardant la ligne d'aide qui s'affiche en bas pour chaque bouton. Cette ligne décrit succinctement l'action de la touche et donne (en rouge au début de la ligne) le raccourci-clavier.

Vous pouvez également taper **xd3d --help** pour afficher la liste des options disponibles au démarrage du programme (très utilisé pour les opérations répétitives, ou la génération de fichiers PostScript en batch).

Les commandes sont classées par type d'action :

1. Géométrie : tout ce qui concerne le cadrage du graphique et l'orientation de l'objet,
2. Présentation : les options de tracé,
3. Fichiers : les commandes qui concernent les fichiers de données,
4. Couleurs : les commandes permettant d'agir sur les couleurs,
5. Sauvegardes : les commandes de sauvegarde sous forme fichier PostScript ou bitmap.

Certaines commandes ont une action immédiate (exemple : tapez **2**. On sauvegarde le dessin sous forme de fichier PostScript noir et blanc et on revient à l'état précédent), d'autres nécessitent la réponse à quelques questions dans le shell initial (qui vient alors en général se placer devant la fenêtre graphique). La plupart des commandes à action immédiate agissent “en bascule” ou de façon “circulaire” : exemple : **i** permet de modifier la couleur des arêtes tracées. Au bout de 16 fois on revient à la couleur initiale. Pour les commandes de ce type, une information parlante est généralement affichée dans la fenêtre de gauche, sur le bouton correspondant à la commande, afin de permettre de s'y retrouver (par exemple pour la commande **i**, on affiche le numéro de la couleur actuelle des arêtes dans cette couleur). Pour les commandes à bascule (par exemple **face cachée**), la touche reste “enfoncée” pour l'un des deux états. Pour les commandes permettant d'atteindre plus de deux états, il y a généralement un menu déroulant. Certaines touches correspondent à des commandes différentes suivant que l'on trace un maillage ou une surface tridimensionnelle. Seule l'action effectivement disponible est indiquée dans la fenêtre de gauche.

4.2 Les commandes

- **Bouton gauche de la souris** : zoom. Cliquez dans la zone graphique. Bougez la souris, un cadre jaune apparaît montrant la nouvelle zone cadrée. Lorsqu'elle vous convient, cliquez à nouveau. On peut abandonner entre les deux clics en tapant `q`.
- **Bouton du milieu de la souris** : translation. Maintenez le bouton enfoncé. Un parallélépipède (ou un rectangle en 2D) matérialisant les bornes de l'objet apparaît. Bougez-le et relâchez le bouton lorsque sa position vous convient.
- **Bouton droit de la souris** : rotation. Maintenez le bouton enfoncé, le parallélépipède apparaît. Bougez le curseur, il tourne ! Attention aux maillages 2D : pour revenir au cadrage initial, tapez `u`.
- **Touche Ne bougeons plus** : elle permet “d'empiler” les commandes sans redessiner à chaque fois. Lorsqu'elle est enfoncée et “allumée”, le dessin n'est pas mis à jour suivant les actions demandées mais les commandes sont prises en compte. Le dessin apparaît après une nouvelle pression sur cette touche. Cette option est utile pour les gros fichiers et/ou les petites machines.
- **Touche reload** : lorsqu'elle est active elle s'allume en rouge. Dans ce cas, l'un des fichiers (maillage ou données) ayant servi à tracer le graphique a été modifié sur le disque depuis sa lecture. On peut le recharger en pressant ladite touche (utile lorsqu'un calcul tourne en tâche de fond pour suivre en temps réel sa progression). Attention, l'accès à un fichier pendant qu'un autre programme est en train d'écrire dedans peut provoquer le décet immédiat d'`xd3d`.
- `0` : zoom (cf. bouton gauche de la souris).
- `1` : symétrise le dessin par rapport aux axes $0x$ et $0y$, uniquement si l'objet initial se trouve dans le “bon” quart d'espace. Au bout de 4 fois on revient au dessin initial.
- `2` : création d'un fichier PostScript noir et blanc (avec niveaux de gris). Ce fichier ne contient que la fenêtre principale — celle du milieu — (le défaut si le maillage seul est affiché) ou bien également les informations de la fenêtre de droite (le défaut en présence d'isovaleurs), selon ce qui est affiché en face de la commande `s` (`cadre` ou `no cadre` en bas à gauche). Un cadre pointillé clignotant et multicolore indique pendant la sauvegarde la zone qui est sauvée. Le début du fichier PostScript contient des informations intéressantes sur la façon dont il a été généré. Vous pouvez les lire directement dans un éditeur ou faire `grep "%>" d3dNN.ps`.
- `3` : couleur du fond. Initialement noir (blanc en PostScript).
- `4` : shrink. Applique à chaque élément une homothétie de rapport 0.8 par rapport à son centre. Permet de vérifier la conformité d'un maillage.
- `5` : translation. Le centre du nouveau graphique se trouve où était le curseur au moment de la commande.
- `6` : élimination des faces cachées ou non (bascule).
- `7` : rotation autour de l'axe $0x$ de l'angle courant (cf. option `b` pour modifier cet angle). Attention, l'axe $0x$ n'est pas celui représenté sur la figure (sauf initialement), mais un axe fixe “dirigé” vers le coin inférieur gauche.
- `8` : rotation autour de l'axe $0z$ de l'angle courant. Attention, l'axe $0z$ n'est pas celui représenté sur la figure (sauf initialement), mais un axe fixe “dirigé” vers le haut de la figure.
- `9` : rotation autour de l'axe $0y$ de l'angle courant. Attention, l'axe $0y$ n'est pas celui représenté sur la figure (sauf initialement), mais un axe fixe “dirigé” vers le coin inférieur droit.

- **a** : zoom arrière de coefficient 1.4.
- **b** : changement de l'angle courant pour les options 7, 8, 9 et **w**.
- **c** : zoom avant de coefficient 1.4.
- **d** : cette touche a une action différente suivant que l'on visualise un maillage ou une surface tridimensionnelle :
 - maillage : lecture d'un fichier contenant un champ scalaire pour le tracé d'isovaleurs ou vectoriel pour le tracé de flèches (cf. paragraphe 3.3). Dans ce dernier cas, un champ scalaire correspondant à la norme de ce champ vectoriel est automatiquement calculé. Pour tracer les flèches associées à un champ vectoriel en même temps que les isovaleurs d'un autre champ scalaire, il faut donc d'abord lire le champ vectoriel, puis le champ scalaire. Pour un maillage 2D, un fichier scalaire lu peut servir à tracer des isovaleurs sur le maillage, ou bien à déformer le maillage suivant Oz pour le visualiser sous forme d'une nappe 3D. On peut ainsi visualiser deux informations simultanément en lisant successivement deux fichiers scalaire et en affectant un aux isovaleurs et l'autre à la déformation suivant Oz . En 2D comme en 3D, un fichier vectoriel peut servir à déformer le maillage (option **n**).
 - surface : normalisation suivant les axes de coordonnées. Transforme le parallélépipède $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \times [z_{\min}, z_{\max}]$ en un cube (en bascule, retour aux proportions réelles).
- **e** : format de la fenêtre graphique : carrée ou rectangulaire (en bascule).
- **f** : si les données possèdent de telles informations, on trace avec des couleurs différentes les arêtes de frontière référencées ainsi que les différents sous-domaines. Les frontières entre les différents sous-domaines sont indiquées en trait rouge épais. On indique également, pour les fichiers `.avoirnD` les blocages des degrés de liberté.
- **g** : légendes sur le dessin. La première fois on entre une légende. Elle s'affiche en haut à gauche du cadre. Les pressions suivantes sur **g** la déplacent aux 4 coins puis au centre. Une pression supplémentaire et elle disparaît. Encore une et il faut écrire une nouvelle légende. Les accents cédilles etc. sont accessibles avec la même syntaxe qu'en $\text{T}_{\text{E}}\text{X}$: `\'e` pour é, `\cc` pour ç etc... (le dernier est un mauvais exemple pour voir si ça suit. Personne n'a jamais écrit de cédille comme ça en $\text{T}_{\text{E}}\text{X}$).
- **h** : affichage des isovaleurs (uniquement si un fichier de scalaires a déjà été lu). Si le fichier contient des valeurs aux noeuds, on affiche successivement des isovaleurs avec remplissage, remplissage+surlignage des frontières des zones de couleur en noir, rajout des valeurs, isovaleurs au trait, isovaleurs au trait+valeurs, et retour à l'état initial. Si les valeurs sont données aux centres des éléments, le programme effectue l'interpolation pour le tracé lissé mais permet aussi d'afficher des valeurs constantes par élément.
- **i** : couleur des arêtes. Permutation sur les 16 couleurs de base.
- **j** : changement des bornes de calcul des isovaleurs.
- **k** : modification de la table de couleurs courante. 16 tables différentes sont disponibles ainsi que leurs inverses (32 tables) :
 - 1 : noir – bleu – vert – jaune – rouge – blanc (table initiale, jolie à l'écran, passe moins bien à l'impression)
 - 3 : bleu – vert – jaune – rouge (celle qui rend le mieux sur les imprimantes couleur)
 - 5 : bleu – mauve – rouge – jaune – blanc
 - 7 : noir – gris – blanc (recommandée pour les impressions à niveaux de gris)
 - 9 : noir – rouge
 - 11 : noir – vert
 - 13 : noir – bleu

- 15 : noir – jaune
- 17 : noir – magenta
- 19 : noir – cyan
- 21 : noir – rouge – blanc
- 23 : noir – vert – blanc
- 25 : noir – bleu – blanc
- 27 : noir – jaune – blanc
- 29 : noir – magenta – blanc
- 31 : noir – cyan – blanc

Les tables $2n$ sont les tables $2n - 1$ inversées. A la vingtième pression on retrouve la table initiale.

- **l** : changement du nombre d'isovaleurs (222 maximum).
 - **m** : affichage des axes de coordonnées. Initialement ils sont dans le coin inférieur gauche. A la première pression ils disparaissent, puis réapparaissent dans le coin supérieur gauche. Ensuite ils tournent autour de la fenêtre pour revenir à leur point de départ.
 - **n** : exagération. Pour les maillages d'éléments finis, cette option n'est active que si le fichier `.depl` a été lu (cf. paragraphe 3.3). Elle sert alors à exagérer les déplacements. Pour les surfaces 3D (fichiers `.c3d` ou courbes paramétrées tapées au clavier), elle permet de définir 3 rapports d'aspect suivant les trois axes de coordonnées afin d'obtenir une figure de proportions plus agréables.
 - **o** : création d'un fichier PostScript couleur (cf. option 2 pour voir ce qui est sauvé).
 - **p** : définition de la couleur de remplissage des faces en l'absence d'isovaleur. Par défaut, les objets 2D sont coloriés en noir (qui devient blanc dans les fichiers PostScript) et les objets 3D simulent l'éclairage par une source directionnelle ("couleur" entre 96 et 99).
 - **q** : sortie du programme.
 - **r** : affichage d'isovaleurs associées aux champs scalaires de chacune des coordonnées des noeuds. Le champ affiché est indiqué dans la fenêtre de droite en dessous du nom du fichier contenant le maillage.
 - **s** : définition des paramètres de la sauvegarde PostScript :
 - signature (sign) : présence en bas du PostScript d'informations concernant l'utilisateur, la date de sauvegarde etc. A enlever pour insérer le dessin dans un document \TeX ou \LaTeX
 - informations (info) : sauvegarde du dessin (fenêtre centrale) et du cadre de droite (échelle de couleurs, informations diverses).
 - orientation (land/port) : "landscape" ou "portrait".
 - dessin seul : sauvegarde du seul dessin (orientation portrait). Pas d'informations à droite, pas d'axes, pas de cadre autour du dessin. Se méfier dans ce mode des flèches éventuellement tracées qui peuvent sortir du cadrage.
 - table seule : sauvegarde du seul cadre de droite (orientation portrait).
- Appuyez successivement sur **s** pour faire apparaître la configuration qui vous convient.
- **t** : cadrage optimal : l'objet remplit toute la fenêtre.
 - **u** : cadrage initial la première fois, puis, en boucle, orientation suivant des axes normalisés.
 - **v** : inversion du point de vue : on se place de l'autre côté de l'objet.
 - **w** : rotation autour d'un axe perpendiculaire à la figure (axe de vision), de l'angle courant (cf. option **b**).
 - **x** : affichage des frontières du domaine en traits plus épais.
 - **y** : hauteur de la feuille (pour PostScript). En bascule 23.5 cm (défaut) et 27.7 cm.
 - **z** : changement de fichier. Il est possible de partir des paramètres initiaux par défaut ou de

- garder les paramètres courants (en particulier la position de l'objet par rapport aux axes).
- ^ : affichage des numéros. La première fois on affiche les numéros des noeuds, puis des éléments, enfin les numéros de références. Le bouton `num` n'a pas exactement le même effet puisqu'il permet d'activer le mode d'affichage des numéros sous le curseur (un à la fois). Pour utiliser ce mode, suivre les instructions sur la ligne d'aide.
- / : permet de définir la façon d'afficher (ou de ne pas afficher) les axes et les échelles.
- . : option active uniquement en présence de plusieurs sous-domaines (maillages) ou plusieurs surfaces (surfaces 3D). Elle permet d'afficher un seul sous-domaine (ou une seule surface), ou tous les sous-domaines moins un.
- - : suppression du tracé du maillage.
-) : comparaison de deux fichiers de valeurs scalaires. On affiche les isovaleurs de l'écart absolu ou relatif des valeurs lues.
- (: affichage ou non des informations du cadre en haut à droite (en bascule).
- ' : lorsqu'un champ scalaire est présent, on affiche sur le graphique la valeur de celui-ci au point indiqué par le curseur.
- % : type de perspective. Successivement :
 - perspective moyenne (le défaut pour le 3D)
 - perspective lointaine
 - perspective isométrique (le défaut pour le 2D)
 - perspective proche
- \$: couleurs imposées. Redéfinition manuelle d'une ou plusieurs couleurs de la table courante pour repérer aisément une isovaleur particulière.
- & : direction des lumières : la source de lumière (visible uniquement en 3D) tourne autour de l'objet (6 positions différentes).
- + : affichage des flèches (après l'option `d`).
- ! :
 - maillages : si le maillage est tridimensionnel, constitué de tétraèdres P1 (à 4 noeuds), d'hexaèdres Q1 (8 noeuds) ou Q2 (27 noeuds), et si un fichier scalaire a été lu au préalable, on trace une isosurface du champ scalaire. On peut tracer l'isosurface seule ou tous les points où le champ scalaire est `j` ou `l` à une valeur donnée. Pour la voir, il faut tracer le maillage sans coloriage grâce à l'option `6` et il est préférable de supprimer le maillage (option `-`) ou de ne garder que son contour géométrique (option `x`). L'option `p` permet de choisir de lisser ou non la surface (réponses entre 96 et 99) et l'option `&` agit sur l'éclairage de l'objet.
 - surfaces 3D : coupes de la surface parallèlement aux axes et appel de `xgraphic`.
- { : positionne l'objet suivant une position angulaire donnée, lue par exemple dans un fichier PostScript généré précédemment et récupérée comme indiqué dans l'option `2`.
- * : facteur multiplicatif de la taille des flèches.
- | : sauvegarde du dessin sous un format bitmap de taille donnée par l'utilisateur. Les formats accessibles sont les suivant (à condition d'avoir installé les différents logiciels de traduction, cf. paragraphe 2) :
 - `xwd` : dump X. Toujours disponible, sauf en cas de mauvaise installation.
 - `ppm` : "portable pixmap" : il sert à obtenir tous les autres. Programme nécessaire : `xwd2ppm`.
 - `tiff` : "Tag Image File Format" compressé ou non. Programmes nécessaires : `xwd2ppm` et `ppmtotiff`.
 - `gif` : "Graphic Interchange Format" entrelacé ou non. Programmes nécessaires : `xwd2ppm` et `ppmtogif`.

- **pict** : format d'image Macintosh. Attention il peut être utile de reconstituer ensuite les ressources Mac du fichier si on veut qu'il soit reconnu sans problème par toutes les applications (cf. le Macintosh-gourou le plus proche de chez vous). Programmes nécessaires : **xwd2ppm** et **ppmtopict**.
 - **rgb** : crée trois fichiers **.red**, **.grn** et **.blu**. Programmes nécessaires : **xwd2ppm** et **ppmtorgb3**.
 - **xpm** : X pixmap. Programmes nécessaires : **xwd2ppm** et **ppmtxpm**.
- Les formats **gif**, **tiff** compressé et **pict** sont beaucoup plus compacts que les autres. Il est également possible de compresser directement les images générées par **gzip**, ce qui permet de gagner beaucoup de place sur les fichiers **xwd**, **ppm**, **rgb** et **xpm**.
- On peut sauver en format bitmap une série d'images correspondant à différentes positions de l'objet ou différents fichiers scalaires associés en vue d'une animation. Par exemple le package **Imagemagick** ou le programme **xanim**, tous deux dans le domaine public, permettent d'animer une série de **.gif**.
- **_** : affichage sur le dessin du titre du travail contenu dans le fichier "avoir" ou "nopo".

5 Annexe. Le format “avoir”

Le petit programme Fortran suivant écrit un fichier de format “avoir” supposé être contenu en mémoire centrale. Il décrit ainsi la structure de données reconnue par xd3d.

```
      real*4  xyz(3,numnp)
      integer info(9,numnp),nbel(numsd),nbnod(numsd)
      integer noeud(nbnodmax,nbelmax)
      character*80 titre
c
c ndim = 2 ou 3 (dimension de l'espace)
c
      if (ndim.eq.2) then
         open(1,file='toto.avoir2D',form='unformatted',recl=112)
      else
         open(1,file='toto.avoir3D',form='unformatted',recl=112)
      endif
c
c titre (chaîne de caractères de longueur 80)
c
      write(1) titre
c
c numnp = nombre de noeuds du maillage
c
      write(1) numnp,8
c
c xyz  = les coordonnées des noeuds
c info = informations diverses
c      ce tableau peut être rempli de données arbitraire
c      (le fichier reste alors utilisable). Les seules
c      informations réellement utiles pour le trace sont~:
c          info(4,n) = numéro de référence du noeud n
c          info(5,n) = numéro de matériau du noeud n
c boucle sur les noeuds
c si ndim = 2, la troisième composante xyz(3,n) doit être constante
c
      do n=1,numnp
         write(1) (xyz(i,n),i=1,3),(info(k,n),k=1,9)
      enddo
c
c numsd      = nombre de sous-domaines
c nbel(nsd)  = nombre d'éléments du sous-domaine nsd
c nbnod(nsd) = nombre de noeuds des éléments du sous-domaine nsd
c les nombres de noeuds acceptés sont~:
c      en 2D : 3 ou 4 (triangles ou quadrangles)
c      en 3D : 3 (surfaces), 4 (tétraèdres), 8 ou 27 (hexaèdres)
c boucle sur les sous-domaines
c
      ne = 0
      do nsd=1,numsd
         write(1) nsd,1,nbel(nsd),nbnod(nsd)
c
c boucle sur les éléments du sous-domaine nsd
c
         do n=1,nbel(nsd)
c
c ne          = numéro global de l'élément
c noeud(i,ne) = noeud numéro i de l'élément ne
c
            ne = ne+1
            write(1) n,(noeud(i,ne),i=1,nbnod(nsd))
         enddo
      enddo
      close(1)
end
```

Pour les allergiques au Fortran, voici un petit exemple en C qui fait à peu près la même chose :

```
extern int cont_m2_sommets, cont_m2_faces;
extern double **m2_sommets;
extern int **m2_faces;

void m2_out(int n_raf, char *nom, int s_flag) {
    char nfout[MAXNOM2];
    FILE *fp;
    int i, j, k, t[4], zero[9], cnt, sint;
    char hed[90];
    float som[3];

    sprintf(nfout, "%s%3d.avoir3D", nom, n_raf);
    fp=fopen(nfout, "wb");
    if (fp == (FILE *)NULL) {
        perror("Cpseudo");
        exit(3);
    }
    sprintf(hed, "%11s %6d%71c", "RAFFINEMENT", n_raf, ' ');
    cnt=80;
    sint=sizeof(int);
    fwrite(&cnt, sint, 1, fp);
    fwrite(hed, sizeof(char), 80, fp);
    fwrite(&cnt, sint, 1, fp);
    t[0]=cont_m2_sommets;
    t[1]=8;
    cnt=2*sint;
    fwrite(&cnt, sint, 1, fp);
    fwrite((char *)t, sint, 2, fp);
    fwrite(&cnt, sint, 1, fp);

    for (j=0; j<9; j++)
        zero[j]=0;
    cnt=3*sizeof(float)+9*sint;
    for (i=1; i<=cont_m2_sommets; i++) {
        for (j=1; j<=3; j++)
            som[j-1]=(float)(m2_sommets[i][j]);
        fwrite(&cnt, sint, 1, fp);
        fwrite((char *)som, sizeof(float), 3, fp);
        fwrite((char *)zero, sint, 9, fp);
        fwrite(&cnt, sint, 1, fp)
    }
    t[0]=1;
    t[1]=1;
    t[2]=nelem_m2;
    t[3]=3;
    cnt=4*sint;
    fwrite(&cnt, sint, 1, fp);
    fwrite((char *)t, sint, 4, fp);
    fwrite(&cnt, sint, 1, fp);
    cnt=4*sint;
    for (i=1; i<=cont_m2_faces; i++) {
        for (j=1; j<=3; j++)
            t[j-1]=m2_faces[i][j];
        fwrite(&cnt, sint, 1, fp);
        fwrite(&i, sint, 1, fp);
        fwrite((char *)t, sint, 3, fp);
        fwrite(&cnt, sint, 1, fp);
    }
    fclose(fp);
} /* Fin de m2_out */
```