

Réaliser des figures en T_EX

par Yvon Lafranche¹

IRMAR, Université de Rennes 1

Campus de Beaulieu, 35042 Rennes Cedex, France.

Yvon.Lafranche@univ-rennes1.fr

Résumé

Fig4T_EX est une petite bibliothèque de macros permettant de réaliser des figures annotées dans un environnement T_EX. Après avoir présenté le contexte dans lequel elle a été développée, on fait un tour d’horizon des possibilités qu’elle offre en mettant en évidence quelques qualités qui en font un outil en général apprécié de ses utilisateurs.

1 Motivation

La création de figures dans un environnement T_EX² est une question qui a été abordée par la communauté des utilisateurs de T_EX il y a déjà plusieurs années et que l’on peut considérer comme résolue. Pourtant des interrogations à ce sujet reviennent de manière récurrente, émanant en particulier de nouveaux utilisateurs de T_EX, ou de plus habitués qui sont confrontés à une situation nouvelle ou à une limitation de la méthode qu’ils utilisent habituellement. Fig4T_EX est l’une des solutions à envisager.

Au fil du temps, plusieurs “packages” ont vu le jour pour répondre aux besoins exprimés. On peut distinguer deux types de solutions : d’une part l’inclusion d’un fichier graphique externe, c’est à dire réalisé par un logiciel graphique dédié, d’autre part la création d’un dessin à l’aide de commandes “embarquées” dans le document T_EX. Fig4T_EX propose ces deux solutions. La première est relativement banale et l’utilisateur dispose d’une assez large panoplie de possibilités pour y parvenir. La seconde l’est moins et nous allons dans ce qui suit nous attacher à décrire ce que Fig4T_EX propose en la matière.

Tout système possède ses particularités, Fig4T_EX n’y fait pas exception. Il a été conçu dans l’esprit de pouvoir reproduire dans un document T_EX le croquis que

1. SMAI-SIGMA

2. ou L^AT_EX, mais le problème étant le même, on écrit T_EX pour simplifier là où il n’y a pas lieu de faire de distinction.

l'on fait sur un coin de table pour illustrer son propos, mettant ainsi en pratique l'adage qui dit qu'un dessin vaut mieux qu'un long discours. Dans le même temps, un autre objectif majeur a été de s'affranchir de toute dépendance vis à vis d'un logiciel autre que T_EX. Cet aspect a guidé la conception du système et influe aussi sur son utilisation.

En effet, en général, on ne se contente pas de faire un dessin. Il est souvent nécessaire de l'annoter, d'y inscrire des légendes. A ce sujet, trois points sont à considérer : l'allure du dessin finalement choisie, la localisation de l'annotation sur le dessin et la police de caractères à utiliser pour l'écrire. Dans la mesure où la création du dessin fait partie intégrante du document, le choix de la police est sans problème : il suffit de procéder comme on le fait normalement au sein du texte en utilisant les commandes de T_EX. La question de la localisation est plus délicate : lorsqu'on fait un dessin on aimerait qu'une légende puisse être associée de manière unique à un élément graphique et le "suivre" en cas de modification. Enfin, la configuration géométrique finale du dessin, pour qu'elle révèle pleinement ses qualités illustratives, est quelquefois longue à mettre au point.

Tous ceux qui se sont confrontés à la réalisation d'une figure le savent bien, on obtient rarement du premier coup le résultat espéré, parce que la configuration géométrique est trop particulière ou pas assez claire ou ne laisse pas de place pour la légende. . . On est donc amené à modifier certains éléments du dessin, corriger la position d'une légende, etc., pour tenter d'obtenir ce que l'on souhaite.

Fig4T_EX est un système suffisamment flexible pour prendre en compte ces questions de façon relativement aisée. Sa conception a été guidée par une *approche géométrique*, ce qui en fait un outil bien adapté aux constructions géométriques. A contrario, il n'est pas vraiment fait pour tracer le graphe de telle ou telle fonction connue de manière analytique. Il existe toutes sortes de logiciels qui font cela très bien. Il suffit ensuite d'inclure dans son document le fichier graphique ainsi créé, sur lequel on peut ajouter des légendes si on le souhaite. Justement, Fig4T_EX est aussi prévu pour cela.

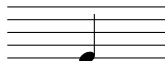
2 Le principe

T_EX permet de tracer des traits verticaux ou horizontaux, ce qui est utilisé pour composer des tableaux. Il permet aussi de matérialiser des points comme ceci \cdot ou comme cela \bullet , mais ne permet pas, de façon native, de tracer des traits obliques ou des courbes, ni de matérialiser des points plus exotiques comme celui-ci \bullet , en supposant que ce soit utile. Cependant, T_EX a été conçu par un esprit ouvert qui a prévu des outils qui autorisent ce genre d'extension. Fig4T_EX exploite ces possi-

bilités et permet ainsi d'élaborer des constructions géométriques dans le plan ou dans l'espace.

L'instrument de base est le point ; il peut être défini par des coordonnées ou obtenu par transformation géométrique. On peut ainsi définir des segments de droites, les regrouper et y adjoindre d'autres éléments graphiques existants com-

me ceci, par exemple :



Cette réalisation nécessite de "connaître un peu la musique". C'est ce que nous allons maintenant nous employer à découvrir au travers de quelques exemples commentés, en nous attachant aux principes plutôt qu'en donnant une liste rébarbative de fonctionnalités.

Le processus de fabrication d'une figure se déroule généralement en trois étapes :

1. définition des points caractéristiques de la construction géométrique,
2. définition des traits du dessin (ils sont enregistrés dans un fichier),
3. composition de la figure par assemblage du dessin et d'une légende éventuelle.

Il reste ensuite à afficher la figure selon la mise en page souhaitée à l'aide des commandes standard de T_EX.

Pour réaliser les étapes décrites ci-dessus, l'utilisateur doit faire usage de commandes particulières, ou macros, selon la terminologie de T_EX. Elles commencent toutes par `\fig`. On peut les classer en trois catégories principales :

- les macros qui commencent par `\figdraw` dont le rôle est de produire les traits du dessin ; elles ne sont utilisées qu'au cours de l'étape 2 précédente ;
- les macros qui commencent par `\figwrite` dont le rôle est d'écrire sur le dessin ; elles ne sont utilisées qu'au cours de l'étape 3 précédente ;
- toutes les autres macros, parmi lesquelles on peut distinguer les macros géométriques et les macros de contrôle.

3 Un petit exemple complet

Inclus quelque part dans un document, le texte suivant dessine le triangle de la figure 1 :

```
% 1. Définition des points caractéristiques
\figinit{pt}
\figpt 2:(-10, -10)
\figpt 3:(110, 10.5)
```

```

\figpt 1:(70, 90) % 3 points dans le plan
\figptbary 5:$G[1,2,3 ; 1,1,1] % Isobarycentre
% 2. Création du fichier graphique
\figdrawbegin{}
\figdrawline [1,2,3,1] % Dessin du triangle
\figdrawend
% 3. Composition de la figure avec écriture des légendes
\figvisu{\figBoxA}{FIGURE 1: Premier exemple.}{
\figwriten 1:(4)
\figwritew 2:(2)
\figwritee 3:(2) % Ecriture aux sommets
\figset write(mark=$\figBullet$) % Changement de marqueur
\figwrites 5:(4) % Ecriture au c.g.
}
% 4. Affichage de la figure
\centerline{\box\figBoxA}

```

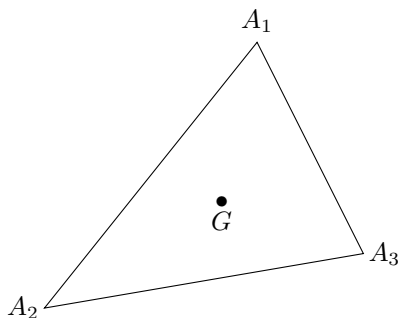


FIGURE 1: Premier exemple.

Pour obtenir ce résultat, il faut ajouter `\usepackage{fig4tex}` juste avant l'instruction `\begin{document}` s'il s'agit d'un document \LaTeX . Il faut ajouter `\input fig4tex` en tête de fichier s'il s'agit d'un document \TeX .

Commentons rapidement ce petit programme.

L'étape 1 commence par une commande d'initialisation qu'il vaut mieux considérer comme obligatoire. L'argument, ici `pt`, précise dans quelle unité l'utilisateur va travailler. Ensuite, quatre points sont définis : les trois sommets du triangle et son centre de gravité. On remarque que les coordonnées sont fournies sans préciser d'unité. L'ordre de définition des trois premiers points est quelconque. Par contre, ils doivent l'être avant la définition du point numéro 5 puisque le calcul de celui-ci en dépend. La commande peut se lire ainsi : calculer le point 5, de texte associé G , barycentre des points 1, 2 et 3, affectés des coefficients 1, 1 et 1.

L'étape 2 regroupe les instructions qui définissent les traits du dessin. Celles-ci doivent figurer obligatoirement entre les deux macros `\figdrawbegin` et `\figdrawend`. Ici, une seule commande suffit pour tracer le triangle, sous forme d'une ligne brisée fermée passant par les sommets.

L'étape 3 est réduite à l'appel à la macro `\figvisu` qui compose la figure et l'enregistre dans une boîte T_EX dont le nom est fourni comme premier argument, ici `\figBoxA`. Le deuxième argument de `\figvisu` est le titre de la figure ; il peut être vide. Le troisième argument peut être vide aussi. Il contient les instructions qui permettent l'inscription des légendes sur le dessin. On reconnaît les macros de type `\figwrite`. La première se lit : écrire au nord du point 1 le texte qui lui est associé par défaut (A_1) à une distance de 4 pt. La transposition est aisée pour les points 2 (écriture à l'ouest) et 3 (écriture à l'est). Par défaut, les points ne sont pas matérialisés. C'est pourquoi, grâce à la commande `\figset`, le marqueur de type bullet est choisi pour être inscrit et ainsi identifier la position du point 5.

L'étape 4 consiste à afficher la figure en vidant le contenu de la boîte construite par `\figvisu`.

Insistons à présent sur la flexibilité du système en faisant observer que, si par exemple la position du point 2 ne convient pas, il suffit de corriger ses coordonnées et de relancer la compilation du document pour obtenir la nouvelle figure, ceci sans rien changer par ailleurs.

Ajoutons quelques remarques techniques complémentaires :

- Les unités autorisées sont celles prévues par T_EX ; les plus couramment utilisées sont le point T_EX (pt), le millimètre (mm), le centimètre (cm), et le pouce (in). On rappelle que $1 \text{ in} = 72.27 \text{ pt}$. L'unité choisie peut être précédée par un facteur d'échelle, de sorte que si la première instruction est changée par exemple en `\figinit{0.6pt}`, la taille de la figure obtenue sera 60% de celle de la figure actuelle. Ce paramètre n'affecte que les coordonnées des points et nullement l'épaisseur des traits ou les polices de caractères. Ceci est donc utile pour calibrer jusqu'au dernier moment la dimension du dessin en fonction de la place disponible dans le document.
- Les points créés sont repérés par un entier positif ou nul choisi par l'utilisateur. Ils portent tous un texte associé qui peut être affiché au cours de l'étape 3 par une macro de type `\figwrite`. Par défaut, le texte associé au point i est A_i . Le texte affiché pourrait aussi être spécifiée lors de l'appel à `\figwrite` ; dans ce cas, il aurait priorité sur le texte associé lors de la définition du point. Le texte associé par défaut peut lui-même être changé par la commande `\figset`, par exemple si l'on souhaite que les points s'appellent P_i . Mais ce texte associé est a priori

quelconque et peut ne pas être un simple nom. Au besoin, ce peut être une boîte de contenu arbitraire.

- Un point peut être redéfini à tout moment. Il est possible, et souvent nécessaire, de créer des points au cours des deux dernières étapes (voir l'exemple suivant).
- L'argument de `\figdrawbegin` est vide la plupart du temps, ce qui correspond au mode de fonctionnement automatique. Dans ce mode, `\figvisu` insère automatiquement le dernier fichier (temporaire) créé au cours de l'étape 2. Lorsque l'argument est fourni, il désigne le nom du fichier dans lequel le dessin est mémorisé, ce qui peut être utile dans certaines circonstances.
- L'affichage de la figure est fait ici à l'aide de la commande `\centerline` communément utilisée en plain T_EX. En L^AT_EX, il est bien sûr possible d'utiliser un environnement dédié tel que `figure` qui permet notamment la numérotation automatique des figures :

```
\begin{figure} [h]
  \centerline{\box\figBoxA}
  \caption{Premier exemple.}
\end{figure}
```

L'instruction `\centerline{\box\figBoxA}` peut dans tous les cas être remplacée par :

```
\begin{center}\begin{tabular}{c}
  \box\figBoxA
\end{tabular}\end{center}
```

- Il existe deux autres boîtes préservées `\figBoxB` et `\figBoxC` utilisables comme argument de `\figvisu` à la place de `\figBoxA`. Elles permettent de faire des assemblages horizontaux de trois figures, ce qui suffit aux besoins les plus courants.
- Les macros que l'on utilise dans FigT_EX sont dites macros à balises. Par défaut, les arguments d'une macro T_EX apparaissent entre `{}`. Ici, des caractères particuliers séparent les arguments. Cette syntaxe est censée rendre leur utilisation plus naturelle et mnémotechnique. Par exemple, la macro de définition d'un point a pour prototypes (versions 2D et 3D) :

```
\figpt NewPt :Text (X, Y)
\figpt NewPt :Text (X, Y, Z)
```

Le caractère `:` sépare le numéro du point du texte associé, et les coordonnées doivent être données entre parenthèses, séparées par une virgule, comme on le fait habituellement.

4 Pour aller plus loin

Nous avons déjà rencontré la macro `\figptbary` qui permet de calculer le barycentre de plusieurs points. A l'usage, on se rend compte que cette macro est très souvent utile pour définir des points qui dépendent d'autres points. Dans cet esprit, FigT_EX propose en outre une série de macros pour effectuer des transformations de points. Le prototype général est

```
\figptXXX Image :Texte= Antécédent /définition de la transformation/
```

où XXX vaut `inv`, `hom`, `rot`, `sym`, `tra` ou `map`, selon le choix de la transformation, à savoir dans l'ordre, l'inversion, l'homothétie, la rotation, la symétrie, la translation ou l'application "générale" définie par une matrice. Pour chacune de ces macros, il existe une version multi-points qui permet de calculer les images de plusieurs antécédents ; leur nom est de la forme `\figptsXXX`.

D'autres macros existent pour calculer des projections ou des lieux géométriques tels que des intersections, ou des points appartenant à des objets géométriques particuliers comme des ellipses ou des courbes de Bézier.

Nous allons en rencontrer quelques-unes au cours des paragraphes suivants.

4.1 Encore des segments de droites

Considérons le programme ci-dessous et la figure qu'il produit :

```
% 1. Définition des points caractéristiques
\figinit{cm}
\figpt 0:$O$(0,0) % Origine
\def\VTx{101} % Nom du
\figvectC \VTx(2,0) % vecteur de translation en x
\def\VTy{102} % Nom du
\figvectC \VTy(0,3) % vecteur de translation en y
\figpt 1:(1,1) % Point inférieur gauche
\figpttra 2:=1/1,\VTx/ % Point inférieur droit
\figptstra 3 = 2,1 /1, \VTy/ % 2 points supérieurs
% 2. Création du fichier graphique
\figdrawbegin{}
% Tracé des axes à l'origine
\figdrawaxes 0(1)
% Droite de symétrie [5,6]
\figpttra 5:=3 /1,\VTx/
\figptbary 6:[5,2 ; -1,3]
\figdrawline[5,6]
% Rectangle initial
\figset (width=0.8)
```

```

\figdrawline[1,2,3,4,1]
% Rectangle symétrique par rapport à la droite [5,6]
\figptssym 11=1,2,3,4 /5,6/
\figset (dash=2)\figdrawline[11,12,13,14,11]
% Rotation autour du point 14, alias C sur le dessin
\figptsrot 21=11,12,13 /14,-130/
\figset (dash=4)\figdrawline[21,22,23,14,21]
\figdrawend
% 3. Composition de la figure avec écriture des légendes
\figvisu{\figBoxA}{FIGURE 2: Transformations.}{
\def\DistAff{2pt}%
\figwritesw 0:(\DistAff)
\figwritew 1,4:(\DistAff)
\figwritee 3:(\DistAff)
\figwritenw 2:(\DistAff)
\figwritee 5:{$(\Delta)$}(\DistAff)
\figwrites 14:$C$(3pt)
}

```

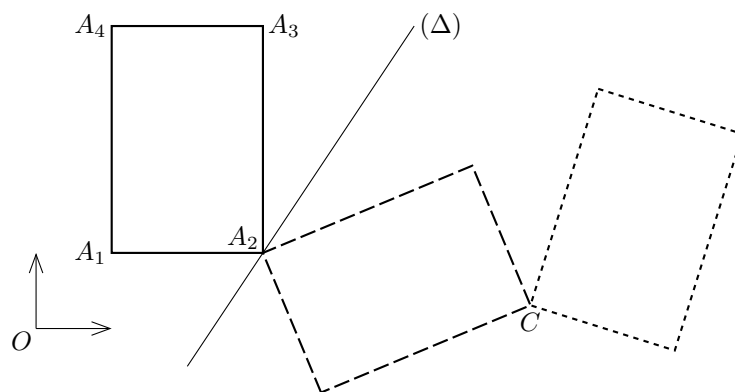


FIGURE 2: Transformations.

L'unité est cette fois le `cm`. Tout comme un point, un vecteur porte un numéro. Les numéros des deux vecteurs de translation sont associés aux noms `\VTx` et `\VTy`³. Les vecteurs sont ensuite définis par la macro `\figvectC` en donnant leurs composantes (ce que signifie `C`). L'intérêt de nommer un vecteur est de pouvoir le référencer dans la suite de façon plus pratique et lisible. Ceci est aussi valable pour des points.

Une fois les coordonnées du point 1 fixées, les trois autres sommets du rectangle (points 2, 3 et 4) sont construits à l'aide de ces translations (1 fois `\VTx` et 1 fois

3. La macro T_EX `\def` est utilisée ici, mais en L^AT_EX, on pourrait utiliser aussi `\newcommand`.

`\VTy`) de sorte que les seules données réelles sont les coordonnées du point 1 et les dimensions du rectangle qui apparaissent dans la définition des vecteurs de translation.

Arrêtons-nous un instant sur l’instruction `\figptstra 3 = 2,1 /1, \VTy/`. Il s’agit d’une commande qui calcule plusieurs points, ce qui est signalé par `pts` dans son nom. Les antécédents sont pris dans l’ordre où ils apparaissent dans la liste d’appel. Les images portent automatiquement les numéros croissants à partir du numéro fourni (ici 3). Cette instruction calcule donc les images 3 et 4 des points 2 et 1 respectivement par la translation de vecteur 1 fois `\VTy`.

Dans l’étape 2, on commence par tracer un petit repère à l’origine dont les flèches sont de longueur 1 (cm). On choisit ensuite de tracer la droite (Δ) passant par le point 2 et parallèle à la diagonale définie par les points 1 et 3. Cette droite est définie par les points 5 et 6 et seul ce segment est tracé.

Jusqu’à présent les traits ont été tracés avec l’épaisseur par défaut. La commande `\figset (width=0.8)` augmente cette valeur à 0.8 bp. Dans ce contexte particulier, l’unité est toujours donnée implicitement en bp, “big point” ou point PostScript, dont la taille est très proche du pt de T_EX. Ce changement est actif jusqu’au prochain changement d’épaisseur éventuel. Ici, tous les traits suivants sont tracés avec cette épaisseur.

Après avoir tracé le rectangle initial, on calcule l’image de ses sommets par la symétrie axiale par rapport à la droite (Δ). Ceci est fait par l’instruction `\figptssym 11=1,2,3,4 /5,6/` qui se lit : calculer les points 11, 12, 13, 14, images respectives des points 1, 2, 3, 4 par la symétrie par rapport à la droite définie par les points 5 et 6. Le rectangle issu de cette symétrie est ensuite tracé après que le type de trait a été changé. Dix types de traits sont prédéfinis ; on peut aussi par la même syntaxe définir un autre motif de pointillé.

Enfin, ce rectangle subit une rotation d’angle -130 degrés autour de l’un de ses sommets, le point 14. Les trois autres sommets sont calculés à l’aide de la macro `\figptsrot` selon le même principe que précédemment pour la symétrie. Le rectangle final est tracé après un dernier changement de type de trait.

Dans l’étape 3, les légendes sont inscrites sur le dessin. Cet exemple apporte quelques nouveautés. On a vu que pour placer un texte par rapport à son point d’attache, on utilise une macro `\figwriteX` où X désigné un des quatre points cardinaux n, s, w, e. Les points cardinaux intermédiaires ne, nw, sw, se sont aussi disponibles. Il s’avère que ces huit directions sont parfois insuffisantes ; c’est pourquoi il existe d’autres macros qui permettent un positionnement très précis.

4.2 Enfin des courbes

L'outil fondamental pour tracer des courbes quelconques est la courbe de Bézier. Fig4T_EX procure une macro pour tracer une chaîne continue de N arcs de courbes de Bézier cubiques dont le prototype est :

```
\figdrawBezier N [Pt_1, ..., Pt_{3N+1}]
```

Le $k^{\text{ème}}$ arc est défini par les quatre points de contrôle $P_{3k-2}, P_{3k-1}, P_{3k}, P_{3k+1}$. Ainsi dans l'exemple suivant, une fois les points de contrôle fixés, le tracé de la courbe s'obtient par `\figdrawBezier 2 [1, 2, 3, 4, 5, 6, 7]`. Il y a un raccord C^1 au point 4 du fait qu'il est le milieu du segment [3,5].

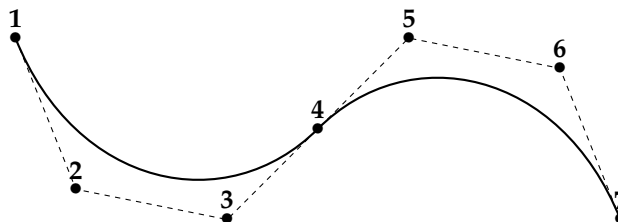


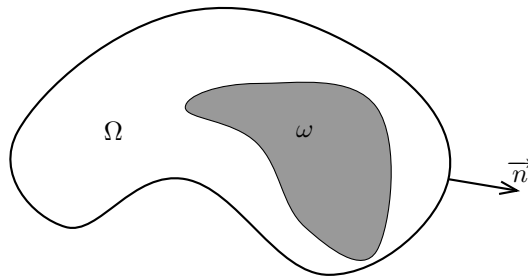
FIGURE 3: Courbes de Bézier.

Il est souvent pratique de procéder par interpolation de points donnés. La macro

```
\figdrawcurve [Pt0,Pt1, ..., PtN,Pt_{N+1}]
```

réalise cela. L'exemple classique est celui du "domaine Ω ", dont la forme typique est le haricot sec, maintes et maintes fois dessiné par nombre d'orateurs! On montre sur la figure 4 un tel domaine Ω contenant un sous-domaine ω . Chaque domaine est dessiné par un appel à `\figdrawcurve`, une fois les points d'interpolation choisis (il y en a ici 7 pour chacune d'elles). La courbe obtenue est de classe C^1 par construction et il est possible d'influer sur sa forme par un paramètre de "tension". L'intérieur d'une courbe peut être colorié par une couleur de son choix ou un gris (ceci est vrai aussi pour une ligne polygonale). Cela s'obtient en écrivant `\figset (fillmode=yes,color=0.6)`, en l'occurrence dans cet exemple, juste avant l'instruction `\figdrawcurve` (ou tout autre commande de tracé). Enfin, il est aisé d'adjoindre à un domaine une normale extérieure en un point quelconque de son contour.

Les macros précédentes permettent de réaliser des formes libres. On est aussi souvent amené à dessiner des formes prescrites comme des ellipses ou des cercles.

FIGURE 4: Un domaine Ω et un sous-domaine ω .

L'exemple du porisme de Steiner fait intervenir des cercles. Il n'est pas spécialement intéressant à ce titre, mais plutôt en ce qu'il "oblige" à établir une véritable construction géométrique permettant son illustration. Le but est de tracer les cercles de droite. Ils sont les images des cercles de gauche par une inversion de centre Ω . Les cercles de gauche sont régulièrement répartis et sont donc plus aisés à obtenir. En plus de transformations déjà évoquées, la procédure mise en œuvre ici utilise la macro `\figptsinterlinell` qui calcule les points d'intersection d'une droite et d'une ellipse. Chaque cercle est tracé par appel à la macro dédiée `\figdrawcirc` qui prend pour données le centre et le rayon du cercle.

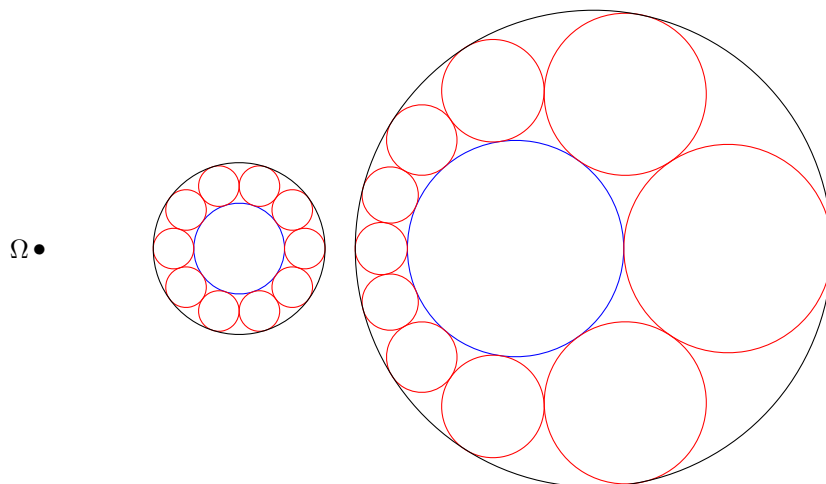


FIGURE 5: Porisme de Steiner.

4.3 La troisième dimension

Un certain nombre de macros sont utilisables en dimension 3 avec la même syntaxe qu'en dimension 2. Pour certaines autres, il suffit d'ajouter un paramètre qui est soit un point, soit un vecteur. Enfin quelques macros sont spécifiques à une dimension donnée. La troisième coordonnée d'un point est 0 si elle est omise.

La procédure générale de création d'un dessin est identique en 2D et en 3D. Cependant, il faut spécifier que l'on souhaite travailler en 3D en ajoutant un argument à `\figinit` : on écrit alors par exemple `\figinit{pt, X}`, où X définit la projection à utiliser et a pour valeur `cavalier`, `orthogonal` ou `realistic`. Comme pour tous les réglages, les paramètres de la projection choisie sont modifiables à l'aide de la macro `\figset`.

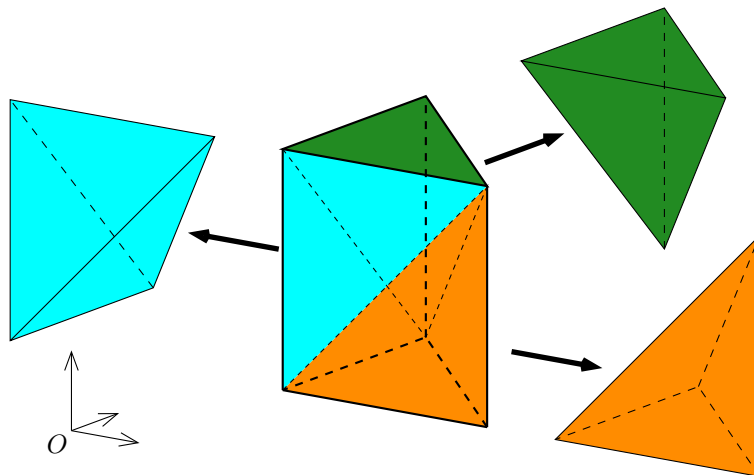


FIGURE 6: Décomposition d'un prisme en tétraèdres.

Il y a six façons de décomposer un prisme en tétraèdres. L'exemple de la figure 6 présente l'une d'entre elles. La projection orthogonale a été sélectionnée en écrivant `\figinit{cm,ortho}`. Les trois sommets de la base du prisme une fois définis ont été translatés par la commande `\figptstra 4=1,2,3/1,100/`, où 100 est le vecteur de translation défini par ses trois composantes à l'aide de la macro déjà rencontrée en 2D : `\figvectC 100(0,0,3)`.

La direction d'observation a été fixée par la commande

```
\figset proj(longitude=-55, latitude=15)
```

où les angles sont donnés en degrés. Cette commande doit naturellement être exécutée avant de commencer l'étape 2.

Les inscriptions sur le dessin se font selon le même principe qu'en 2D, c'est à dire en utilisant le repérage par points cardinaux, mais cette fois autour de la projection du point d'attache du texte.

4.4 Et plus encore...

Il n'est pas question de faire ici une revue exhaustive de toutes les fonctionnalités du système. On aurait pu s'attarder sur les diagrammes, les flèches courbes, les macros spécifiques au triangle ou aux courbes de Bézier, le calcul de distances ou d'angles, les outils de projection en 2D et 3D ou de calcul d'intersections. Seuls certains réglages d'attributs graphiques et non graphiques ont été rencontrés au fil des exemples. Les macros d'écriture spécialisées n'ont été qu'évoquées. Il existe aussi des macros d'aide à la réalisation d'une figure, notamment `\figshowpts` et `\figshowsettings`.

Mieux qu'une telle énumération, l'échantillon d'exemples qui suit, donnés sans commentaire technique, devrait témoigner du spectre des possibilités accessibles avec `Fig4\TeX`.

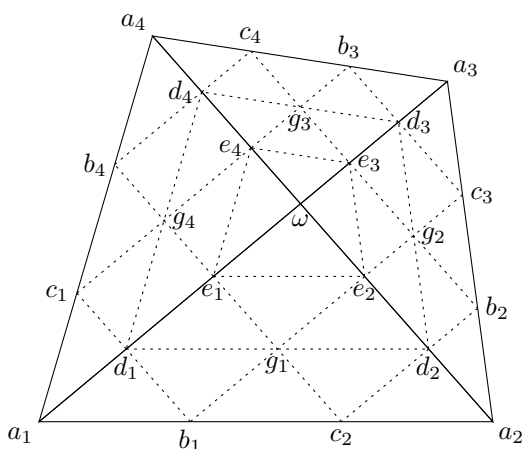


FIGURE 7: Elément FVS.

La figure 7 représente le réseau Bézier associé à l'élément composite de Fraeijs de Veubeke et Sanders. Le quadrilatère est divisé en quatre triangles sur chacun desquels l'interpolant est un polynôme de degré 3 défini par 10 des coefficients $\{\omega, a_i, b_i, c_i, d_i, e_i, g_i, i = 1, 2, 3, 4\}$ conformément aux inscriptions faites sur la figure.

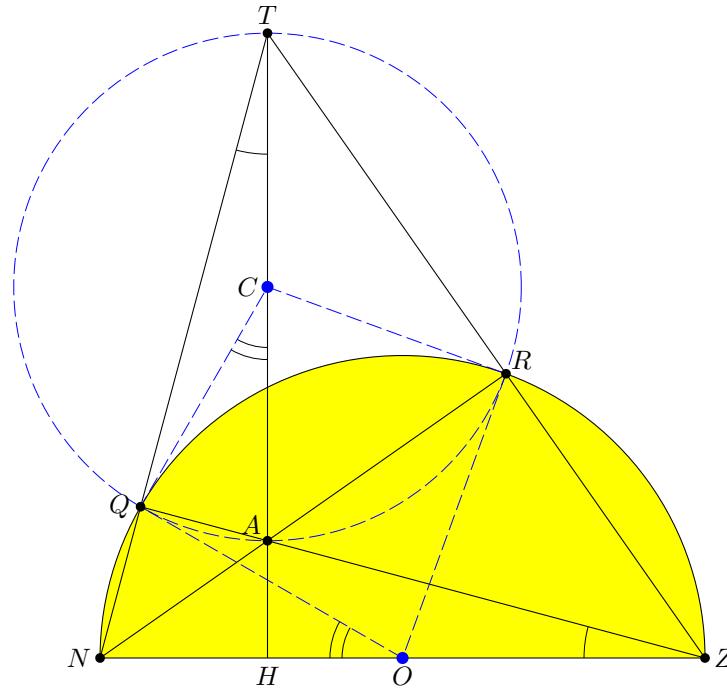


FIGURE 8: Problème du Monde 701.

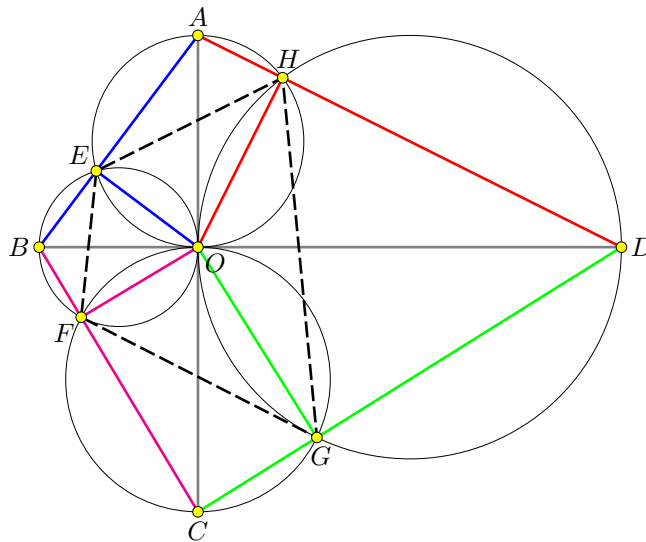


FIGURE 9: Problème du Monde 726.

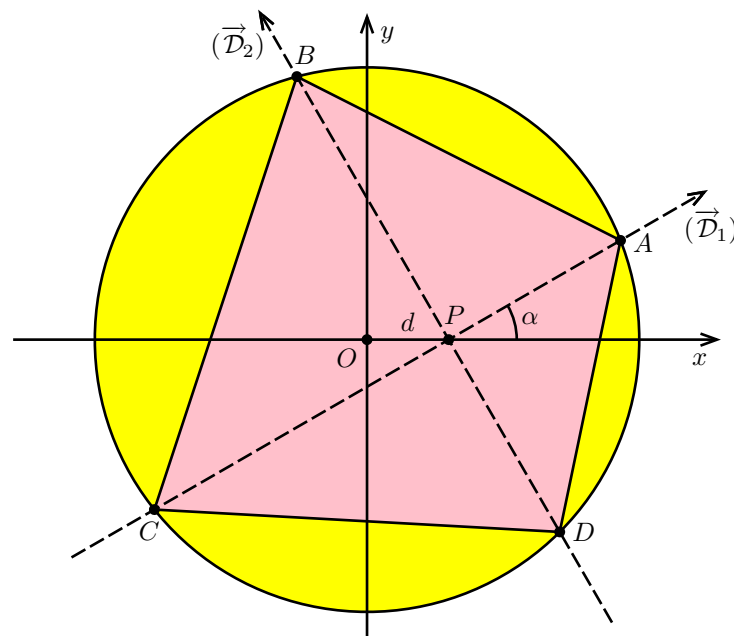


FIGURE 10: Problème du Monde 741.

Les figures 8, 9 et 10 illustrent des propriétés géométriques variées relatives à des contextes spécifiques.

Les deux exemples suivants sont réalisés en dimension 3. Dans les deux cas, la projection “réaliste” a été choisie.

La figure 11 montre le calcul du centre O de la sphère circonscrite au tétraèdre $ABCD$. Le point Ω_c est le centre du cercle circonscrit au triangle BCD qui définit le plan P . Le point O cherché est obtenu par intersection de la droite orthogonale au plan P passant par Ω_c et du plan Q médiateur du segment $[A, B]$.

La figure 12 est censé représenter le schéma d’un dispositif optique. Le point P_i est le point d’impact sur un écran vertical dans la direction α issue de la source S .

5 Conclusion

Fig4T_EX est un ensemble de macros conçues pour *créer* une figure dans un environnement T_EX et pour *écrire* dessus. Ses principales caractéristiques sont :

- Toute l’information pertinente est localisée en un seul endroit, le document T_EX en cours de composition, ce qui simplifie la tâche de l’utilisateur.

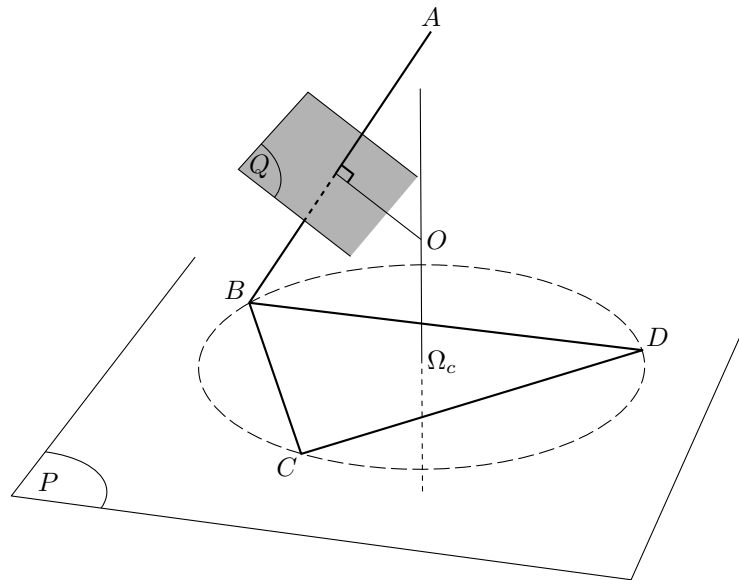


FIGURE 11: Centre de la sphère circonscrite à $ABCD$.

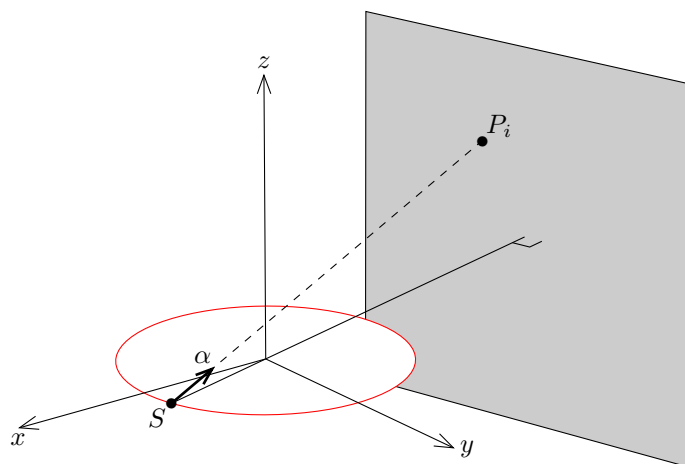


FIGURE 12: Dispositif optique.

- Chaque figure est gérée par T_EX par l’intermédiaire d’une boîte, l’outil fondamental de T_EX, dont les dimensions dépendent automatiquement du dessin et de sa légende éventuelle ; la prise en compte d’une figure s’inscrit donc dans le processus normal de composition du document.
- Fig4T_EX s’adapte automatiquement au mode de composition du document sélectionné par l’utilisateur (mode DVI, compilation par `tex` ou `latex`, ou bien mode PDF, compilation par `pdftex` ou `pdflatex`).
- Les macros de Fig4T_EX sont conçues d’un point de vue géométrique ; une grande variété de constructions géométriques peuvent être réalisées de façon aisée et naturelle à partir de points, de vecteurs et de transformations géométriques. Si ces outils sont utilisés de manière raisonnée, la figure obtenue peut être modifiée en changeant seulement une donnée amont dans le processus de construction, par exemple une coordonnée de point ou la direction d’un vecteur.
- La position de chaque légende inscrite sur le dessin est *liée* à la géométrie. La *même* description géométrique est utilisée à la fois pour tracer la figure et pour positionner la légende. En conséquence, chaque inscription “suit” le dessin si celui-ci est modifié.
- Les constructions géométriques peuvent être réalisées en 2 ou en 3 dimensions ; en 3D, la figure finale est le résultat d’une projection dont la nature et la direction sont paramétrables par l’utilisateur.
- Fig4T_EX est disponible sur le site Web
[http ://perso.univ-rennes1.fr/yvon.lafranche/fig4tex](http://perso.univ-rennes1.fr/yvon.lafranche/fig4tex)
sur lequel on trouve notamment une documentation au format PDF, un manuel de référence en ligne, un guide d’initiation et des exemples.

“Dessine-moi un mouton !”, demande l’enfant à l’aviateur isolé. De tout temps et en toute circonstance, les hommes ont cherché à dessiner pour représenter leur environnement et leurs pensées, et plus généralement pour communiquer entre eux. L’écriture elle-même, de quelque époque ou quelque origine qu’elle soit, est une forme de dessin. Joindre un dessin à un texte pour sa valeur explicative est souvent nécessaire, en particulier dans le domaine scientifique. Fig4T_EX n’existe que pour satisfaire cet objectif. A chacun de juger, en fonction de ses propres besoins, de son utilité.

