

# Manuel utilisateur de DR. GEO



Hilaire Fernandes, Andrea Centomo,  
Jean-Philippe Georget,  
*OFFSET*  
<http://www.offset.org>

26 septembre 2004



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>7</b>
1.1	Avant propos . . . . .	7
1.2	DR. GEO sur le web . . . . .	9
<b>2</b>	<b>Fonctions de base</b>	<b>11</b>
2.1	Outils de construction . . . . .	11
2.1.1	Outils de point . . . . .	12
2.1.2	Outils de ligne . . . . .	13
2.1.3	Outils de transformation . . . . .	14
2.1.4	Outils numériques . . . . .	15
2.1.5	Outil macro-construction . . . . .	16
2.2	Autres fonctions . . . . .	16
2.2.1	Arbre logique de construction . . . . .	16
2.2.2	Déplacer la figure . . . . .	17
2.2.3	Déplacer un objet . . . . .	17
2.2.4	Supprimer un objet . . . . .	17
2.2.5	Changer l'aspect d'un objet . . . . .	17
2.2.6	Changer les propriétés d'un objet . . . . .	20
2.2.7	Afficher une grille . . . . .	21
2.3	Préférences utilisateurs . . . . .	21
2.3.1	Comportement par défaut . . . . .	21
2.3.2	Autres préférences . . . . .	22
<b>3</b>	<b>Fonctionnalités avancées</b>	<b>23</b>
3.1	Macro-construction . . . . .	23
3.1.1	Création d'une macro-construction . . . . .	24
3.1.2	Exécution d'une macro-construction . . . . .	26
3.2	Script Scheme DR. GEO . . . . .	28
3.2.1	DGS par l'exemple . . . . .	28
3.2.2	Méthodes de référence pour les scripts DR. GEO . . . . .	31
3.3	Figure Scheme de DR. GEO . . . . .	34
3.3.1	Quelques exemples . . . . .	34
3.3.2	Méthodes de référence pour les Figures Scheme DR. GEO . . . . .	36
3.3.3	Synonymes des commandes Figure Scheme DR. GEO . . . . .	44
3.3.4	Galerie d'exemples . . . . .	45
3.4	Masquer des outils dans l'interface . . . . .	47
3.4.1	Verrouiller des outils . . . . .	47
3.4.2	Déverrouiller des outils . . . . .	48

<b>4</b>	<b>Fichiers et documents</b>	<b>51</b>
4.1	Enregistrement d'une construction . . . . .	51
4.2	Enregistrement d'une session . . . . .	51
4.3	Enregistrer une macro-construction . . . . .	52
4.4	Ouvrir un fichier . . . . .	52
4.5	Exporter une figure . . . . .	52
4.5.1	Exportation L <sup>A</sup> T <sub>E</sub> X . . . . .	52
4.5.2	Exportation PostScript . . . . .	53
4.5.3	Exportation image PNG . . . . .	53
4.5.4	Exportation Fly Draw . . . . .	53
4.5.5	Définir la zone exporter . . . . .	53
<b>5</b>	<b>Applications didactiques</b>	<b>55</b>
5.1	Pythagore et scripts . . . . .	55
5.2	Théorème et conjectures . . . . .	56
5.3	Nombre irrationnel . . . . .	58
5.4	Spirale de Baravelle . . . . .	60
5.5	Catena di Pappo . . . . .	63
5.6	Calcul de $\pi$ . . . . .	64
<b>6</b>	<b>Astuces diverses</b>	<b>67</b>
6.1	Créer un polygone régulier . . . . .	67
6.1.1	Le modèle de polygone . . . . .	67
6.1.2	La macro-construction du polygone . . . . .	67
6.1.3	Quelques considérations . . . . .	68
6.2	Imprimer une figure . . . . .	68
6.3	Placer un paragraphe de texte dans une figure . . . . .	68
<b>A</b>	<b>Historique</b>	<b>75</b>
<b>B</b>	<b>GNU Free Documentation License</b>	<b>77</b>



# Copyright

Copyright (c) 2000-2004 OFSET.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation ; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Other copyright apply to the respective owners of the modified documentations. See the section history for the complet list.



# Chapitre 1

## Introduction

### 1.1 Avant propos

DR. GEO est un logiciel libre de géométrie interactive et de programmation en Scheme. Il permet de créer des figures géométriques et de les manipuler interactivement en respectant leurs contraintes géométriques. Il offre également la possibilité d'introduire graduellement la programmation. Il est ainsi utilisable dans des situations d'enseignement du niveau primaire au niveau supérieur.

L'interface utilisateur de DR. GEO a été conçue pour allier dans un ensemble harmonieux à la fois simplicité d'utilisation, ergonomie et fonctionnalités avancées.

Ainsi l'interface de DR. GEO, sous une très grande apparence de simplicité, permet au néophyte de se familiariser très rapidement avec les fonctions de base du logiciel. Puis, au fur et à mesure de sa progression, l'utilisateur découvrira des aspects plus avancés de l'interface et du fonctionnement de DR. GEO : multiplicité des modalités de construction d'objet<sup>1</sup>, macro-construction, sessions, adaptabilité de l'interface, scriptabilité, Figure Scheme de DR. GEO. Ces fonctionnalités avancées génèrent peu de surcharge sur l'interface, c'est pour cela que DR. GEO est très agréablement utilisé en enseignement primaire, cependant il est également très intéressant pour le lycée.

Dans les sections suivantes, les outils de base seront exposés. Ensuite les fonctionnalités avancées seront présentées en détail.

L'agencement de l'interface est :

1. la *barre de menu* caractéristique avec **Fichier**, **Édition**, **Macro-constructions**, **Fenêtres**, **Aide** ;
2. la *barre d'action* pour créer une nouvelle figure, un texte d'explication, les outils faire/defaire, etc.

Pour créer une nouvelle figure géométrique, l'utilisateur clique sur le premier bouton de la barre d'action. Alternativement, une nouvelle figure peut être créée en utilisant le menu **Fichier**->**Nouveau**->**Figure**. Lorsqu'une nouvelle figure est créée, une nouvelle *barre d'icônes* avec un ensemble de six icônes apparaît. Cette barre d'icônes se termine par un menu permettant d'ajuster l'échelle de la figure géométrique courante.

En outre, une barre d'icônes verticale de raccourcis, à gauche de la figure, offre un accès rapide aux outils les plus utilisés.

Les six icônes de la barre d'icônes sont des entrées de menu pour des fonctions spécifiques. Ces fonctions sont décrites dans le prochain chapitre.

<sup>1</sup> Il s'agit de pouvoir, à partir d'une même commande, créer un type d'objet selon des modalités différentes. Par exemple à partir de la commande construction de cercle, l'utilisateur peut créer un cercle à partir de son centre et un point, ou bien une longueur, ou bien un segment, etc. Bien sûr cette commande n'est représentée que par une seule icône, il incombe à DR. GEO d'anticiper sur la construction de l'utilisateur. L'effet immédiat est donc une diminution de la charge cognitive de l'interface sur l'utilisateur, tout en proposant un nombre important de modes de construction.

FIG. 1.1 – Écran de bienvenue de DR. GEO

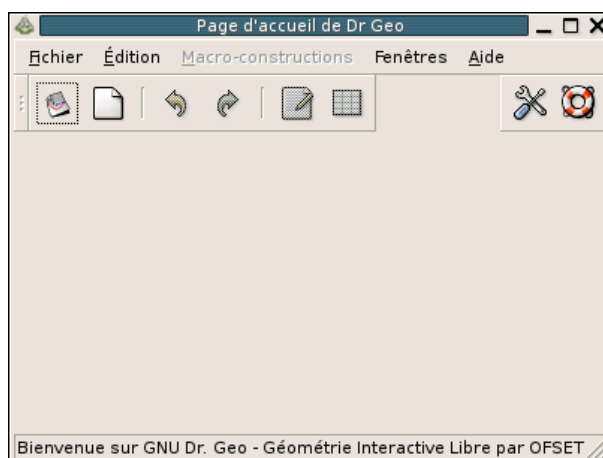
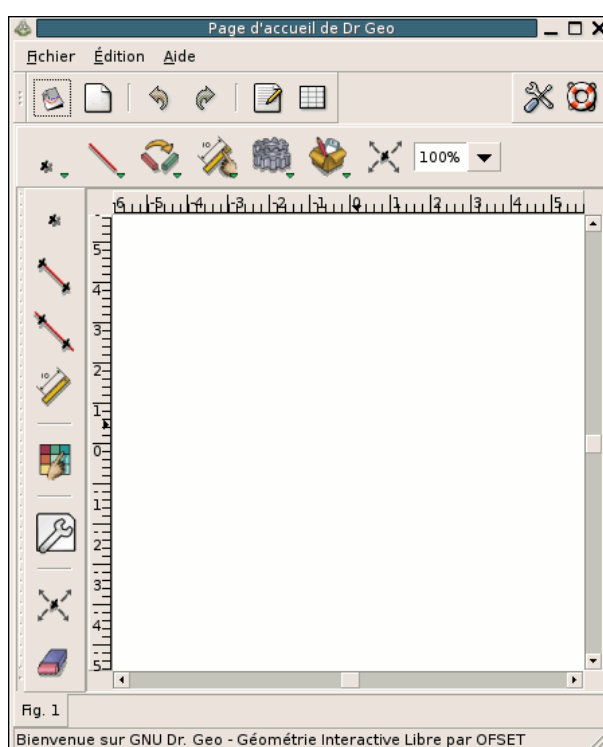


FIG. 1.2 – Une figure géométrique vide

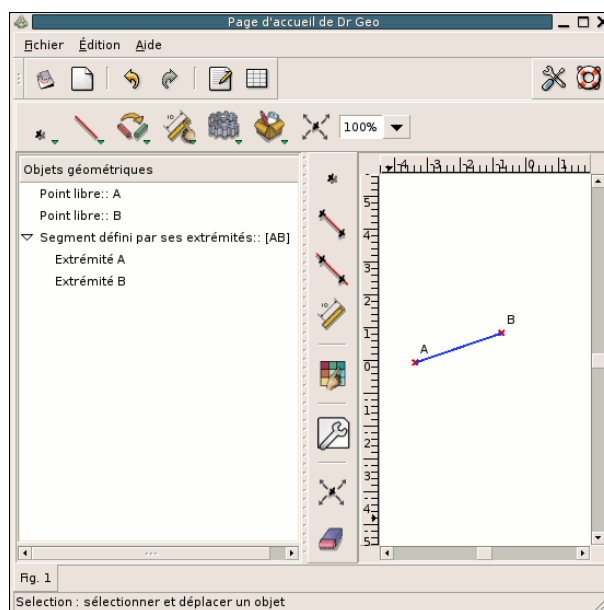


Pour chaque figure, un *panneau synoptique* décrivant sa séquence de construction est disponible. Par défaut ce panneau est poussé à l'extrême gauche et seule la représentation de la figure est visible. À tout moment, l'utilisateur peut pousser le panneau vers la droite pour rendre visible sa description. Une description de figure est un arbre composé de tous les éléments de la figure. Les éléments relatifs à d'autres éléments peuvent être développés en cliquant sur le signe '+', de manière à visualiser les parents de ces éléments.





FIG. 1.3 – Une figure DR. GEO et sa description



## 1.2 Dr. Geo sur le web

DR. GEO dispose de son propre espace web sur le site d'OFSET à l'adresse : <http://www.ofset.org/drgeo>

Sur cet espace, l'utilisateur trouvera les informations suivantes :

- les informations pour obtenir DR. GEO ;
- la documentation sur le logiciel ;
- des indications pour s'impliquer dans le projet DR. GEO ;
- des références sur des exploitations pédagogiques du logiciel.





## Chapitre 2

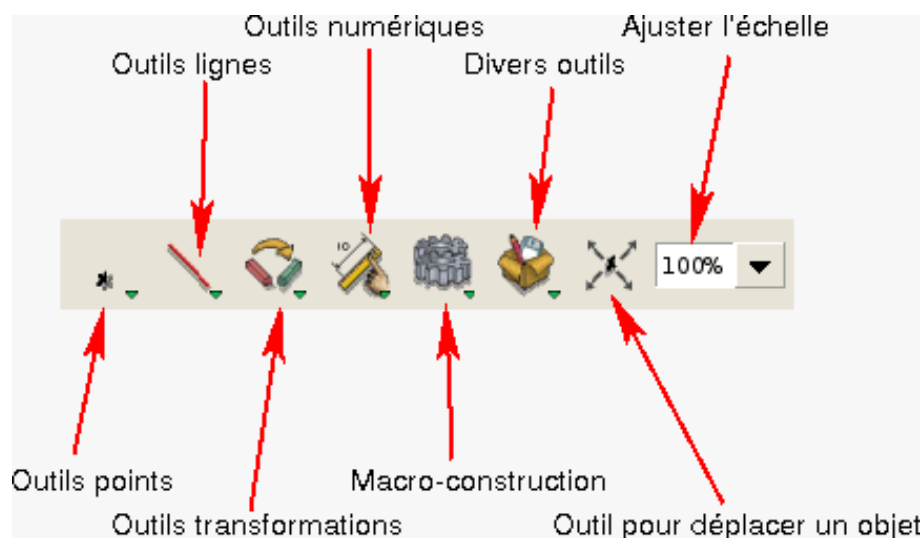
# Fonctions de base

Ce chapitre décrit les outils utilisés pour construire une figure géométrique. Il se termine par la présentation de la configuration des préférences par l'utilisateur.

### 2.1 Outils de construction

Ces outils sont partagés en six groupes accessibles à partir de la seconde barre d'icônes de DR. GEO.

FIG. 2.1 – Catégories d'outils de DR. GEO et leur description



Lorsque l'utilisateur clique sur une des icônes décorées d'un petit triangle vert, une nouvelle barre verticale d'icônes s'affiche immédiatement. Celle-ci regroupe des fonctions d'une même famille.

De la gauche vers la droite, nous avons accès aux barres d'icônes verticales pour construire des points et des lignes, utiliser des transformations, calculer des valeurs, gérer des macro-constructions, et utiliser les outils d'édition – *Divers outils* dans la figure.

Ces fonctions se retrouvent à l'identique dans le menu contextuel d'une figure appelé par un clic droit dans le fond de celle-ci.

### 2.1.1 Outils de point

#### Point libre



Crée un point libre dans le plan ou sur un objet unidimensionnel (segment, demi-droite, droite, arc de cercle, cercle, lieu) :

1. Dans le premier cas, le point créé peut être déplacé n'importe où dans le plan de la figure. Pour le construire l'utilisateur clique simplement sur le fond.
2. Dans le deuxième cas, le point n'est libre que dans l'objet unidimensionnel (ligne) où il a été créé ; il est collé sur l'objet. Pour construire ce type de point, l'utilisateur clique sur une ligne (i.e. droite, demi-droite, segment, cercle, arc de cercle, etc.).

**Comment placer un point avec des coordonnées données ?** Le plus simple est de placer un point libre puis d'éditer ses propriétés – outil Propriétés Section 2.2.6, page 20 – afin d'ajuster ses coordonnées comme souhaité.

Une autre possibilité – moins souple pour ce cas de figure – est de placer deux valeurs libres dans la figure – outils Numériques Section 2.1.4, page 15 – puis de construire le point ayant pour coordonnées ces deux valeurs – outil Point défini par ses coordonnées Section 2.1.1, page 12. Cette possibilité a un avantage sur la précédente, le point ainsi construit ne peut pas être déplacé directement à la souris, le point est en quelque sorte bloqué dans sa position.

#### Milieu



Crée le milieu de deux points ou d'un segment :

1. Dans le premier cas, l'utilisateur sélectionne deux points.
2. Dans le deuxième cas, l'utilisateur sélectionne simplement un segment.

#### Intersection



Crée le ou les points d'intersection de deux lignes (i.e. droite, demi-droite, segment, arc de cercle, cercle). L'utilisateur doit sélectionner deux lignes.

#### Point défini par ses coordonnées



Crée un point défini par ses coordonnées. L'utilisateur doit sélectionner deux nombres, le premier correspond à l'abscisse, le second à l'ordonnée.

**Comment placer un point contraint par ses coordonnées ?** Cette fonction est largement utilisée lorsque nous souhaitons par exemple construire le lieu d'un point. Cette construction suppose au préalable l'existence de deux valeurs – voir Section numérique Section 2.1.4, page 15 – le point est ensuite construit en sélectionnant ces deux valeurs.



### 2.1.2 Outils de ligne

#### Droite



Crée une droite définie par deux points. L'utilisateur sélectionne deux points.

#### Demi-droite



Crée une demi-droite définie par deux points. L'utilisateur sélectionne deux points, le premier est l'origine, le second appartient à la demi-droite.

#### Segment



Crée un segment donné par deux points.

#### Vecteur



Crée un vecteur donné par deux points. L'utilisateur sélectionne deux points, le premier est l'origine, le second est l'extrémité.

Une fois que le vecteur est créé, celui-ci peut être déplacé indépendamment des deux points. Ceci reste vrai pour un vecteur construit par une transformation (Cf. la section Transformations de ce manuel)

#### Cercle



Crée un cercle. L'utilisateur peut créer un cercle à partir de différentes sélections :

1. le centre et un point du cercle ;
2. le centre et un nombre (le rayon du cercle) ;
3. le cercle et un segment dont la longueur est le rayon du cercle.

#### Arc de cercle



Crée un arc de cercle défini par trois points. Le premier est l'origine de l'arc, le troisième est l'extrémité, le second est un point de l'arc.

#### Lieu



Crée un lieu défini par deux points. L'utilisateur sélectionne deux points, l'un des deux est un point sur une ligne, l'autre est un point sous contraintes du premier (i.e. quand l'un bouge, l'autre fait de même).



## Polygone



Crée un polygone défini par  $n$  points. L'utilisateur sélectionne  $n+1$  points limitant le polygone. Le premier et le dernier sont un seul et même point ce qui indique à DR. GEO que la sélection est terminée. L'objet polygone n'est pas un objet comme les autres lignes, il n'est pas possible de placer un point dessus ou de construire une intersection entre un polygone et une autre ligne. En revanche il est possible de construire l'image d'un polygone par une transformation géométrique.

### 2.1.3 Outils de transformation

#### Droite parallèle



Crée une ligne parallèle à une direction et passant par un point. L'utilisateur sélectionne un point et une direction (i.e. une droite, une demi-droite, un segment ou un vecteur).

#### Droite perpendiculaire



Crée une droite perpendiculaire à une direction et passant par un point. L'utilisateur sélectionne un point et une direction (i.e. une droite, une demi-droite, un segment ou un vecteur).

#### Symétrie axiale



Crée l'image d'un objet par une symétrie axiale. L'utilisateur sélectionne l'objet à transformer et l'axe de symétrie (une droite). Quand l'utilisateur veut construire l'image d'une droite, la première droite sélectionnée est la droite à transformer.

#### Symétrie centrale



Crée l'image d'un objet par une symétrie centrale. L'utilisateur sélectionne l'objet à transformer et le centre de symétrie (un point). Quand l'utilisateur veut construire l'image d'un point, le premier point sélectionné est le point à transformer.

#### Translation



Crée l'image d'un objet par une translation. Quand l'utilisateur veut construire l'image d'un vecteur, le premier vecteur sélectionné est le vecteur à traduire.



### Rotation



Crée l'image d'un objet par une rotation. L'utilisateur sélectionne l'objet à transformer, le centre et l'angle de la rotation. Quand l'utilisateur veut créer l'image d'un point, le premier point sélectionné est le point à transformer.

L'angle peut être sélectionné à partir de différents types de valeurs :

- **valeur numérique** : l'angle est alors exprimé en radians. Exemples de valeurs numériques : valeur libre, distance entre deux points, longueur d'un segment, une coordonnée, une valeur retournée par un script Scheme DR. GEO, etc. ;
- **la mesure d'un angle géométrique formé par trois points** : sa mesure est alors exprimée en degrés. Attention, dans ce cas la mesure appartient seulement à l'intervalle  $[0 ; 180]$  ;
- **la mesure d'un angle orienté de deux vecteurs** : sa mesure est exprimée en degrés et couvre l'intervalle  $]-180 ; 180]$ .

### Homothétie



Crée l'image d'un objet par une homothétie. L'utilisateur sélectionne l'objet à transformer, le centre et le facteur (i.e. un nombre). Quand l'utilisateur veut créer l'image d'un point, le premier point sélectionné est le point à transformer.

#### 2.1.4 Outils numériques

##### Distance, longueur & nombre



Crée une valeur numérique. La valeur numérique, selon la sélection de l'utilisateur, peut être calculée ou bien saisie :

1. deux points : distance entre ces deux points ;
2. un segment : la longueur de ce segment ;
3. un vecteur : la norme de ce vecteur ;
4. un cercle : le périmètre de ce cercle ;
5. un arc de cercle : la longueur de cet arc ;
6. une droite : la pente de cette droite ;
7. une droite et un point : la distance entre ce point et la droite ;
8. un **clic souris directement sur le fond de la figure** permet à l'utilisateur d'entrer une nouvelle valeur (i.e. une valeur libre).

Cette dernière possibilité est très intéressante dans certaines situations. Elle permet de fixer une longueur, le rayon d'un cercle, la mesure d'angle (en radian) ou les coordonnées d'un point. La valeur est ensuite utilisée à partir des outils spécifiques de construction de cercle, de rotation et de point défini par ses coordonnées.

### Angle



Calcule la mesure d'un angle défini par trois points ou deux vecteurs. Dans le premier cas, l'angle est considéré comme non orienté (i.e. angle géométrique dont la mesure est dans l'intervalle  $[0; 180]$ ). Dans le second cas, l'angle est orienté et sa mesure est dans l'intervalle  $]-180; 180]$ .

### Coordonnées



Crée les coordonnées (abscisse et ordonnée) d'un point ou d'un vecteur.

### Script Scheme Dr. Geo



Crée un script Scheme DR. GEO. Le script reçoit  $n$  objets en entrée. Il retourne éventuellement un nombre, affiché dans la figure. Un script peut être utilisé pour ses effets de bord ou pour sa valeur de retour. Les scripts Scheme DR. GEO sont couverts en détails dans le chapitre des fonctionnalités avancées Chapitre 3 et exactement à la section script Section 3.2, page 28.

## 2.1.5 Outil macro-construction

### Créer une macro-construction



Extrait une séquence de construction d'une figure et la transforme en macro-construction.

### Exécuter une macro-construction



Exécute (i.e. “lance” ou “joue”) une macro-construction pré-construite. La macro-construction peut être nouvelle ou chargée depuis un fichier.

---

(!) Les macro-constructions sont présentées dans la Section macro-construction 3.1, page 23.

---

## 2.2 Autres fonctions

### 2.2.1 Arbre logique de construction

Chaque figure est associée à un arbre logique de construction. Cet arbre est chronologique, c'est-à-dire qu'il reprend, du haut vers le bas, l'ordre de construction de la figure. Certaines entrées de l'arbre peuvent être dépliées afin de faire apparaître les antécédents – c'est à dire les objets parents – utilisés lors de la définition de l'objet.

Par défaut l'arbre est masqué, il est en fait replié sur le bord gauche de la fenêtre, pour le faire apparaître il faut le déplier à l'aide de la souris. Déplacer le pointeur souris vers le bord gauche de la fenêtre de DR. GEO lorsque celui-ci se transforme en “< – >”, presser le bouton souris et tirer vers la droite.





### 2.2.2 Déplacer la figure

La figure peut être déplacée en appuyant sur la touche “Ctrl” et le premier bouton de la souris.

### 2.2.3 Déplacer un objet



Un objet peut être déplacé par glisser-déposer. La figure est alors redessinée en respectant ses propriétés. Quasiment tous les objets géométriques peuvent être déplacés. Si nécessaire, DR. GEO déplace les points libres associés. Par exemple, quand l'utilisateur déplace une droite définie par deux points, DR. GEO déplace les deux points simultanément.

### 2.2.4 Supprimer un objet



Un objet d'une figure peut être supprimé en activant ce menu. Éventuellement, l'utilisateur peut annuler la suppression à l'aide de la fonction d'annulation à partir de la barre d'icônes ou du menu de l'application. Par défaut le nombre d'annulations possibles est de 10 mais l'utilisateur peut ajuster cette valeur depuis la boîte de dialogue des préférences.

### 2.2.5 Changer l'aspect d'un objet



Chaque objet géométrique a ses propres attributs de style comme la couleur, l'épaisseur, l'étiquette, la taille et la forme. De plus, il est possible de cacher temporairement un objet sans le supprimer. Par exemple, il peut être utile de cacher des constructions intermédiaires sans les supprimer. Tous ces attributs sont ajustables depuis la boîte de dialogue qui s'affiche lorsque l'utilisateur clique sur un objet de la figure. Pour cela il faut d'abord se mettre dans le mode d'édition de style en sélectionnant **Autres->Aspect** du menu contextuel de la figure, ou bien en cliquant sur l'icône ci-dessus, du 6ème tiroir d'icônes.

La boîte de dialogue du style d'un point concerne tous les types de point. Il est possible d'ajuster la couleur, la forme, la taille, le nom et la visibilité.

La boîte de dialogue du style d'une ligne concerne les droites, les demi-droites, les segments, les vecteurs, les cercles, les arcs de cercle, les lieux de points. Il est possible d'ajuster la couleur, le style, le nom et la visibilité. Lorsqu'une droite, une demi-droite, un vecteur ou un segment sont définis par deux points ayant des noms non vides, leur nom est automatiquement déduit à partir de celui des deux points. Dans ce cas, l'utilisateur ne peut pas les renommer.

La boîte de dialogue du style d'un nombre ou d'un polygone concerne toutes les sortes de valeur (saisie par l'utilisateur, calculée par un script Scheme DR. GEO ou représentant une mesure géométrique) et les formes polygonales.



FIG. 2.2 – Boîte de dialogue pour le style d'un objet point

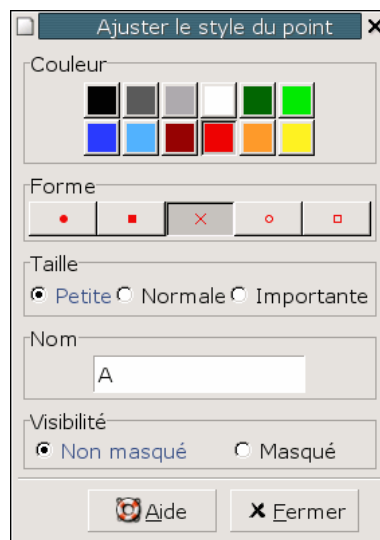


FIG. 2.3 – Boîte de dialogue pour le style d'un objet ligne

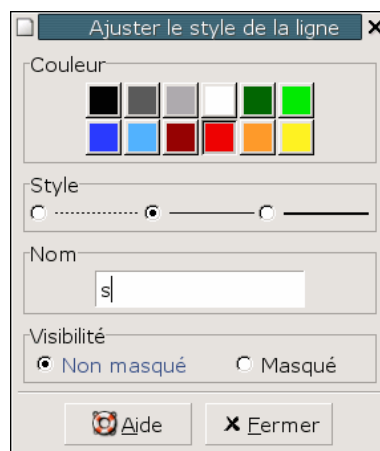
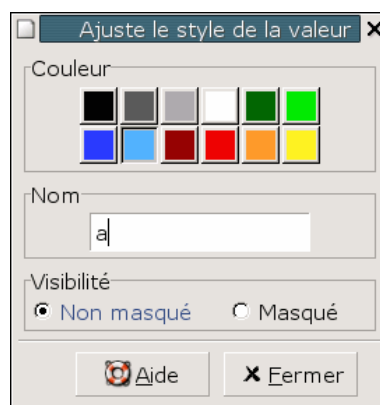


FIG. 2.4 – Boîte de dialogue pour les nombres & les objets polygones



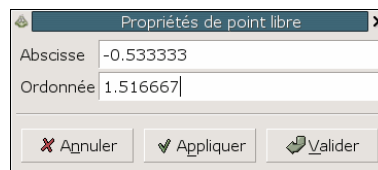
## 2.2.6 Changer les propriétés d'un objet



Certaines propriétés d'objets sont paramétrables par l'utilisateur. Quand l'utilisateur clique sur ces objets, une boîte de dialogue appropriée apparaît. Actuellement, les objets suivants sont concernés :

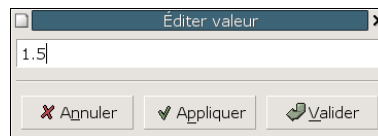
1. point libre : abscisse et ordonnée peuvent être éditées ;

FIG. 2.5 – Changement des coordonnées d'un point libre



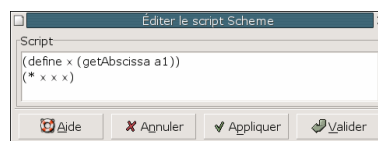
2. valeur libre : sa valeur peut être modifiée ;

FIG. 2.6 – Modification d'une valeur libre



3. script : son code peut être modifié.

FIG. 2.7 – Modification d'un script



### 2.2.7 Afficher une grille

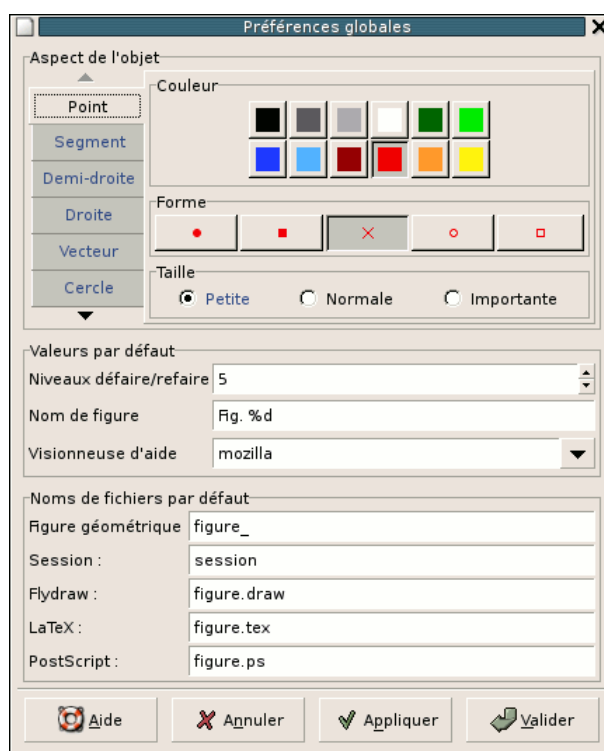
Il est possible d'afficher une grille unitaire dans toute figure de DR. GEO, la commande est accessible depuis le menu **Édition->Montrer ou cacher la grille**. Elle peut également être activée par le raccourci clavier **Ctrl-G**. Si la commande est réactivée, la grille est cachée. La grille est unitaire, chaque subdivision représente une unité. Enfin, lors de la sauvegarde d'une figure, l'état de la grille est également sauvegardé (affichée ou non affichée).

## 2.3 Préférences utilisateurs

### 2.3.1 Comportement par défaut

Le comportement par défaut de DR. GEO peut être configuré de plusieurs manières. Pour paramétrer les préférences, l'utilisateur accède au menu **Édition->Préférences...** pour ouvrir la boîte de dialogue des préférences.

FIG. 2.8 – Préférences des figures géométriques



La boîte de dialogue est composée en deux parties :

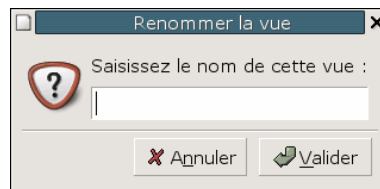
1. La première partie concerne les préférences des figures géométriques. Des onglets permettent à l'utilisateur de paramétrer le réglage par défaut de chaque type d'objet (géométrique ou numérique). Les réglages concernent l'aspect des objets.
2. La deuxième partie concerne les préférences globales :
  - Le nombre de niveaux pour Annuler/Refaire ;
  - Le nom par défaut lorsqu'une nouvelle figure est créée. Le %d est remplacé par un entier géré par DR. GEO, cette valeur est incrémentée à chaque nouvelle figure ;
  - Le navigateur Internet – visionneuse – par défaut pour visualiser l'aide en ligne ;
  - Les noms par défaut utilisés lors de la sauvegarde d'une figure et d'une session ;
  - Les noms par défaut utilisés lors d'exportations aux formats  $\text{\LaTeX}$  et PostScript ;



### 2.3.2 Autres préférences

En plus du comportement par défaut de DR. GEO, l'utilisateur peut modifier le nom d'une figure à partir du menu **Édition->Renommer**.

FIG. 2.9 – Changement du nom d'une figure



## Chapitre 3

# Fonctionnalités avancées

Dans ce chapitre, nous présentons les fonctionnalités qui permettent d'étendre les possibilités de DR. GEO ou de l'adapter à une situation pédagogique donnée.

La première est la *macro-construction*. Elle permet l'extraction d'une construction logique dans un enregistrement. Cet enregistrement peut ensuite être répété ou sauvegardé dans un fichier portant l'extension **.mgeo** et ouvert lorsque nécessaire.

Les scripts Scheme DR. GEO – DR. GEO Script aka DGS – représentent une autre fonctionnalité pour étendre DR. GEO. Ces scripts sont de véritables items de figure, comme les items géométriques. En entrée, ils reçoivent une ou plusieurs références d'items géométriques et ils retournent une valeur placée dans la figure. Ce sont en fait des fonctions<sup>1</sup> greffées dans une figure, elles sont évaluées à chaque mise à jour de la figure (i.e. lorsque la figure a besoin d'être redessinée).

Les scripts Scheme DR. GEO sont utiles pour la valeur qu'ils retournent ou leur effet de bord, cela dépend de ce que l'utilisateur souhaite réaliser.

En extension des scripts Scheme, DR. GEO propose d'aller encore plus loin avec les figures Scheme de DR. GEO. Cette fois il s'agit de décrire une figure géométrique complètement sous la forme d'un code source écrit dans le langage Scheme. La force de cette approche est de permettre une construction fonctionnelle<sup>2</sup> des figures et non plus simplement déclarative comme c'est le cas avec l'interface graphique.

Enfin, l'adaptation de l'interface utilisateur de DR. GEO permet à un enseignant de préparer une session de travail avec des documents dans lesquels certaines fonctions ont été bloquées par un mot de passe. L'intérêt est de contraindre à l'utilisation de certains outils pour des situations pédagogiques données.

### 3.1 Macro-construction

Une macro-construction ressemble un peu à une procédure qui reçoit des items d'une figure en entrée et qui retourne un ou plusieurs items de figure, construits par la macro-construction. Une macro est construite à partir d'un modèle défini par l'utilisateur. Cela signifie que l'utilisateur doit réaliser la séquence de construction une première fois dans une figure puis demander à DR. GEO de l'enregistrer dans une macro-construction. La macro-construction peut ensuite être sauvegardée dans un fichier d'extension **.mgeo**.

Pour enregistrer une séquence de construction, DR. GEO doit connaître les items initiaux de la séquence ainsi que les items finaux. Évidemment, les items finaux ne doivent dépendre *que* des items initiaux<sup>3</sup>, sinon DR. GEO ne sera pas capable de déduire les items finaux à partir des items initiaux.

<sup>1</sup>Ou procédures pour les amateurs de Pascal.

<sup>2</sup>Par exemple sous forme récursive

<sup>3</sup>Cette contrainte a depuis été assouplie et permet d'aller encore plus loin avec les macro-constructions. Voir Créer un polygone régulier Section 6.1, page 67.

Ainsi, DR. GEO déduit la logique de la séquence de construction et l'enregistre dans une macro-construction. L'utilisateur peut exécuter cette macro-construction, elle demande seulement les items initiaux (du bon type) de la figure et construit les items résultants.

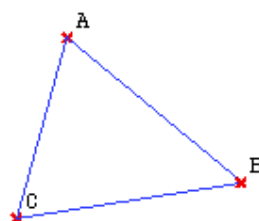
---

(!) Les items de figure intermédiaires et invisibles sont aussi construits par la macro-construction. Ils sont nécessaires pour construire les items résultants.

---

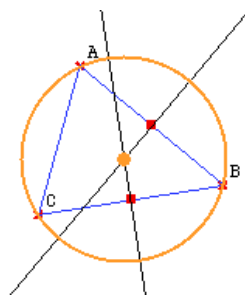
Pour illustrer la fonctionnalité macro-construction, nous utiliserons l'exemple de l'utilisateur qui souhaite enregistrer la construction du cercle passant par trois points ainsi que son centre.

FIG. 3.1 – La figure initiale



Avant la création de la macro-construction, l'utilisateur doit construire la figure finale, elle est considérée comme modèle par la macro-construction.

FIG. 3.2 – La figure avec la construction finale



### 3.1.1 Création d'une macro-construction

À cette étape, la séquence de construction est faite. L'utilisateur doit maintenant avertir DR. GEO qu'il veut une macro-construction à partir de cette séquence. Il doit appeler la

fonction **Construire une macro** à partir de la barre d'icônes



ou depuis le menu contextuel de la vue de la figure.

Depuis la boîte de dialogue de l'assistant, l'utilisateur sélectionne les paramètres d'entrée et de sortie, le nom et la description de la macro-construction.

La seconde page de la boîte de dialogue sert à sélectionner les paramètres d'entrée. Dans notre exemple, ce sont les trois points initiaux. L'utilisateur a juste besoin d'aller jusqu'à cette seconde page et il peut sélectionner les trois points de la figure. Les items sélectionnés clignotent.





FIG. 3.3 – Première page de la boîte de dialogue de l'assistant pour construire une macro-construction

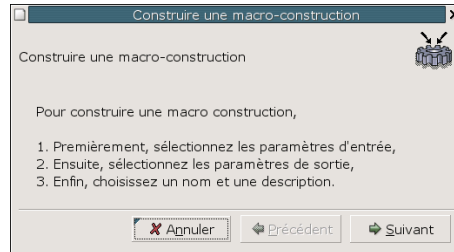
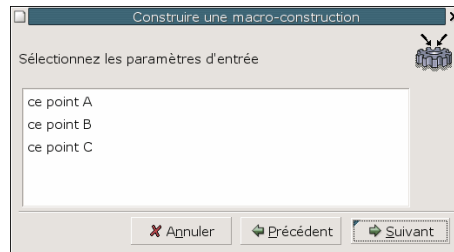
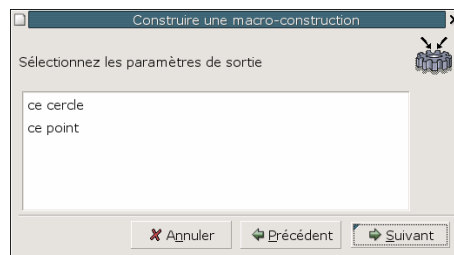


FIG. 3.4 – La seconde page, les trois points sont sélectionnés



À partir de la troisième page, l'utilisateur sélectionne les paramètres de sortie. Dans notre exemple, nous voulons le cercle et son centre comme résultat de la macro-construction. Il procède comme dans le cas des paramètres d'entrée pour les sélectionner.

FIG. 3.5 – La troisième page, le cercle et son centre sont sélectionnés



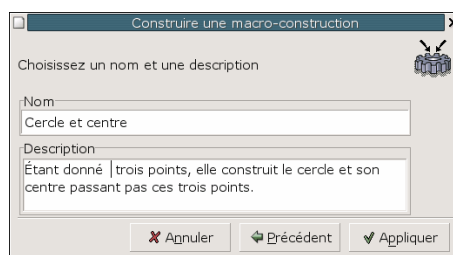
À partir de la quatrième page, l'utilisateur entre le nom et la description de la macro-construction. Ces informations sont affichées lorsque l'utilisateur exécute une macro-construction. Ceci permet de distinguer les macro-constructions entre elles.

À partir de la dernière boîte de dialogue de l'assistant (la cinquième), l'utilisateur termine la création en appuyant sur le bouton **Terminer**. Il peut aussi revenir aux étapes précédentes pour ajuster les paramètres de la macro-construction.

(!) Si la sélection des paramètres d'entrée et de sortie ne correspond pas (DR. GEO ne peut pas extraire la logique de la construction), la macro-construction ne peut pas être créée. Dans ce cas, l'utilisateur doit reconsidérer la sélection des paramètres d'entrée et de sortie. Il peut revenir à la seconde ou la troisième page de la boîte de dialogue de l'assistant pour ajuster ses choix.



FIG. 3.6 – La quatrième page, le nom et la description de la macro-construction



À cette étape, la macro-construction est créée et enregistrée dans DR. GEO. Dans la prochaine section, nous verrons comment l'utiliser.

### 3.1.2 Exécution d'une macro-construction

#### À l'aide de la boîte de dialogue

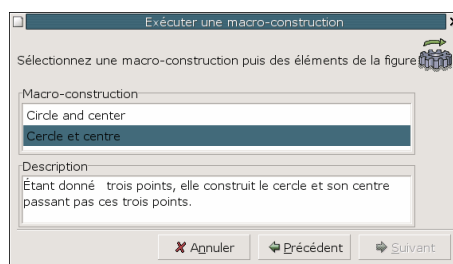
Pour exécuter une macro-construction, l'utilisateur peut appeler la fonction **Exécuter**



une macro préconstruite à partir de la barre d'icônes ou depuis le menu contextuel de la vue de la figure. Une boîte de dialogue décrivant la procédure s'affiche alors.

À partir de celle-ci, l'utilisateur sélectionne la macro-construction. Dans la seconde page, il sélectionne la macro-construction dans la liste en haut de la boîte de dialogue. Une fois la macro sélectionnée, il peut directement cliquer sur les paramètres d'entrée dans la figure. Dès que tous les paramètres d'entrée sont sélectionnés, la macro-construction est exécutée et les paramètres finaux sont affichés.

FIG. 3.7 – L'utilisateur sélectionne les paramètres d'entrée dans la figure



Dans notre exemple, la macro-construction nécessite trois paramètres d'entrée (trois points) et elle construit un point et un cercle. Pour exécuter notre macro-construction, il faut une figure avec au moins trois points.

Une fois que notre macro-construction est appliquée à ces trois points, nous avons le cercle souhaité et son centre.

#### À l'aide du menu Macro-constructions

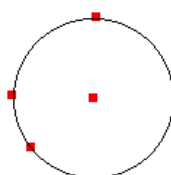
Il existe une autre procédure – plus rapide – pour exécuter une macro-construction. La barre de menu principale de DR. GEO comporte un menu **Macro-constructions**. Ce menu est peuplé par les noms des macro-constructions chargées en mémoire dans le programme. Pour exécuter une macro-construction, l'utilisateur sélectionne directement celle de son choix.



FIG. 3.8 – Une figure avec trois points

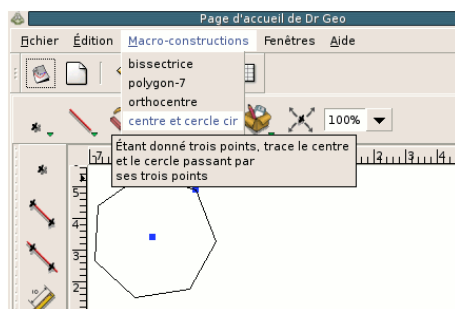


FIG. 3.9 – La figure finale avec le cercle et son centre



En outre un passage du pointeur souris au dessus de chaque item de menu fait apparaître une info-bulle de la description de la macro. L'utilisateur peut ainsi avoir rapidement une explication sur l'ensemble des macro-constructions.

FIG. 3.10 – Exécuter une macro-construction directement depuis le menu Macro-constructions



## 3.2 Script Scheme Dr. Geo

DR. GEO est compatible Guile. Cela signifie qu'il est possible d'exécuter des scripts Scheme avec DR. GEO. Mais qu'est-ce que Guile ? Extrait du manuel de Guile :

*Guile est un interpréteur pour le langage de programmation Scheme, prévu pour être utilisé dans une grande variété d'environnements.*

Les citations suivantes décrivent précisément comment Guile est utilisé dans DR. GEO :

*Comme un shell, Guile peut être lancé de manière interactive, recevoir des expressions de l'utilisateur, les évaluer et afficher les résultats, ou comme un interpréteur de scripts, lire et exécuter du code Scheme à partir d'un fichier. Cependant, Guile est disponible sous forme de bibliothèque permettant à d'autres applications d'incorporer facilement un interpréteur Scheme complet. Une application peut utiliser Guile comme un langage d'extension, un langage de configuration propre et puissant, ou comme une "colle" multi-usages, liant des primitives fournies par l'application.*

Dans DR. GEO, une API est disponible à partir de l'interpréteur Guile. C'est un ensemble de "crochets" dans le moteur géométrique. C'est pourquoi l'utilisateur peut écrire des scripts pour manipuler les items (géométriques ou numériques) des figures. Aussi, puisque les scripts sont des items de figure au même titre que d'autres, ils n'ont pas besoin d'être dans un fichier séparé, ils sont enregistrés dans le fichier de la figure. Dans ce qui suit, nous allons utiliser l'acronyme DGS pour désigner un script Guile DR. GEO ("DR. GEO Script").

### 3.2.1 DGS par l'exemple

L'outil pour créer un DGS est disponible depuis la section numérique du menu contextuel ou de la barre d'icônes. Un DGS peut recevoir de 0 à  $n$  paramètres d'entrée.

Après avoir choisi l'outil, il suffit de cliquer sur les objets souhaités en entrée puis quelque part sur le fond de la figure, le script sera placé à ce dernier endroit.

Dans la suite nous vous proposons de travailler sur quelques exemples de DGS, leurs fonctionnalités et leur puissance seront plus facilement analysées. Les DGS comme les macro-constructions donnent une dimension particulière à DR. GEO, ils permettent – chacun avec un positionnement différent<sup>4</sup> – d'aller là où les auteurs du logiciel ne sont pas allés ou ne souhaitent pas aller.

Il est aussi important de comprendre que la plupart des fonctionnalités de l'interpréteur GNU Guile sont disponibles depuis les DGS. C'est particulièrement vrai pour ses bibliothèques de fonctions<sup>5</sup>, nous allons bien sûr les utiliser intensément.

**DGS sans paramètre d'entrée** La procédure pour créer un script sans paramètre d'entrée est la suivante :

1. Après avoir choisi l'outil de script Section 2.1.4, page 16, cliquez *directement* dans le fond d'écran, à l'emplacement où vous souhaitez placer le script. **Surtout ne cliquez sur aucun objet** de la figure, sinon DR. GEO comprendra que vous souhaitez cet objet comme paramètre d'entrée du script<sup>6</sup>.
2. Une fois le script placé, vous voyez la chaîne de caractères "Dr. Genius" s'afficher. Tout script nouvellement créé contient une commande par défaut pour afficher cette chaîne de caractères, vous pouvez l'éditer en allant à l'outil propriétés d'objet Section 2.2.6, page 20.

<sup>4</sup>Les macro-constructions ont une approche géométrique tandis que les DGS ont une approche numérique mais aussi et surtout nous pouvons les utiliser dans un esprit de bidouillage ("hacking" en anglais).

<sup>5</sup>En particulier, les fonctions mathématiques

<sup>6</sup>Si par accident vous cliquez sur un objet, sélectionnez de nouveau l'outil de script. Cela annulera votre sélection.



3. Une fois cet outil choisi, cliquez sur le script – ou pour être plus précis sur sa valeur – de votre choix dans la figure. Une boîte de dialogue contenant le script s’affichera. Dans la suite de cette section, c’est dans ce dialogue que nous saisissons les scripts.

**Un générateur de nombres aléatoires et autres :** Si vous souhaitez un générateur de nombres aléatoires, rien de plus simple, saisissez pour tout script le code suivant :

```
(random 10)
```

À chaque mise à jour de la figure, il génère un nombre aléatoire entier dans l’intervalle  $[0; 10[$ .

Si vous préférez un nombre flottant dans l’intervalle  $[0; 1[$ , utilisez ce script :

```
(random:uniform)
```

---

(!) Quelques précisions :

- La valeur retournée par le script est la valeur calculée par la dernière ligne. Ici en l’occurrence, il s’agit de la valeur retournée par l’appel d’une fonction ;
  - La dernière ligne d’un script doit retourner un nombre réel. Sinon DR. GEO affiche “Résultat non imprimable” ;
  - Si l’on souhaite retourner la valeur d’une variable, il suffit de mettre son nom en dernière ligne.
- 

**Calculer des valeurs usuelles :** Pour calculer une valeur approchée de  $\pi$  :

```
(acos -1)
```

ou de  $e$  :

```
(exp 1)
```

Les valeurs retournées par ces DGS sont ensuite utilisables comme toutes les autres valeurs numériques que peut générer DR. GEO. Pour toutes ces petites choses les DGS sont donc vos amis. Mais ils peuvent faire bien plus de choses intéressantes lorsqu’ils reçoivent des paramètres en entrée.

**DGS avec au moins un paramètre d’entrée** La procédure pour créer un DGS avec un paramètre d’entrée est sensiblement la même. Juste après avoir sélectionné l’outil script, il suffit de cliquer sur l’objet qui sera passé en paramètre d’entrée, puis de cliquer sur le fond de l’écran, à l’endroit où le script doit être placé.

Ensuite, dans le script, la référence du paramètre d’entrée est placée dans la variable  $a1$ . Si nous avons défini deux paramètres d’entrée, leurs références auraient été placées suivant l’ordre de sélection dans les variables  $a1$  et  $a2$ . De même pour un nombre de 3, 4, etc. de paramètres d’entrée.

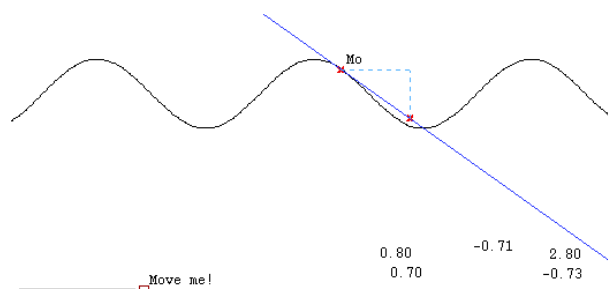
Selon le type d’objet en référence, diverses méthodes sont disponibles qui pour obtenir sa valeur, qui pour obtenir ses coordonnées, etc. Le répertoire des méthodes est disponible depuis la Section Méthodes de référence des DGS 3.2.2, page 31.

Dans la suite, nous exposerons graduellement la construction d’une portion de courbe représentative d’une fonction et la tangente en un point mobile à cette portion de courbe.

La figure finale est fournie dans DR. GEO. Elle s’appelle `slope.fgeo(/usr/share/drgeo/examples/figures/slope.fgeo)`.



FIG. 3.11 – La figure que nous obtiendrons



**Définir une valeur dans un intervalle donné :** Dans une nouvelle figure, nous commençons par placer deux points et le segment dont les extrémités sont ces deux points. Sur ce segment nous plaçons un point libre appelé “Move me!”. Ce point servira de jauge, à nous de la calibrer avec un script. Ensuite nous créons un script ayant pour unique paramètre d’entrée ce point là.

Lorsque  $a1$  est une référence vers le point “Move me!”, le script suivant nous retourne une valeur décimale comprise entre  $[-10; 10]$  :

```
(define x (getAbscissa a1))
(* 20 (- x 0.5))
```

Quelques explications s’imposent. Dans la première ligne, l’appel (`getAbscissa a1`) permet d’obtenir l’abscisse curviligne de l’objet référencé par  $a1$ <sup>7</sup> – c’est-à-dire le point “Move me!”. Cette abscisse est placée dans une nouvelle variable  $x$ . Ensuite à la deuxième ligne nous calibrons, ici l’expression est équivalente en notation algébrique à  $20 \times (x - 0.5)$ . Comme c’est la dernière ligne du script le résultat de ce calcul est la valeur retournée et affichée dans la figure.

Enfin, nommons ce script  $Xo$ .

**Afficher une portion de courbe représentative d’une fonction :** La valeur obtenue par le script précédent nous servira d’abscisse, nous allons nous servir de cette valeur pour calculer avec un deuxième script l’ordonnée en ce point par la fonction  $x \rightarrow \cos(x)$  :

```
(define x (getValue a1))
(cos x)
```

L’appel (`getValue a1`) permet d’obtenir la valeur numérique de l’objet de référence  $a1$ . Ici il s’agit d’obtenir la valeur retournée par le script précédent. Nous nommons ce script  $Yo$ .

Ensuite nous créons le point  $Mo$  de coordonnées  $(Xo; Yo)$ . C’est un point de la courbe représentative de  $x \rightarrow \cos(x)$ . Pour afficher sa portion de courbe représentative sur l’intervalle  $[-10; 10]$ , nous créons le lieu du point  $Mo$  lorsque le point “Move Me!” décrit le segment. Ça y est nous avons une courbe!

**Calculer et afficher la tangente à une courbe :** Pour afficher la tangente en  $Mo$ , il nous faut d’abord la pente en ce point. Nous avons donc besoin de la fonction dérivée  $x \rightarrow -\sin(x)$  et de  $Xo$ . Nous créons un script avec comme paramètre d’entrée le script  $Xo$  :

```
(- 0 (sin (getValue a1)))
```

<sup>7</sup>Cette abscisse est dans l’intervalle  $[0; 1]$ , quel que soit le type de ligne



La notation préfixée utilisée par Scheme/Guile peut dérouter, c'est simplement une question d'habitude. Nommons ce script "pente en Mo". Ainsi lorsque Mo se déplace sur la courbe, la pente est recalculée.

Il nous reste maintenant à afficher la tangente. Pour ce faire nous allons d'abord calculer les coordonnées d'un deuxième point – M1 – de cette droite.

Commençons par son abscisse, par exemple  $X1 = X0 + 2$ , pour ce faire créons un script avec comme paramètre d'entrée le script X0 :

```
(define x1 (getValue a1))
(+ x1 2)
```

Nommons ce script X1.

Attaquons-nous maintenant à l'ordonnée de M1. Là nous avons besoin de :

- Mo (référence *a1*) ;
- la pente en Mo (*a2*) ;
- l'abscisse X1 (*a3*).

Dans le script suivant, nous calculons l'ordonnée de M1 par le calcul  $Y0 + m \times (X1 - X0)$  :

```
(define x0 (car (getCoordinates a1)))
(define y0 (cadr (getCoordinates a1)))
(define m (getValue a2))
(define x1 (getValue a3))
(+ (* m (- x1 x0)) y0)
```

Quelques mots sur l'appel (`getCoordinates a1`), dans cet appel *a1* doit être une référence vers un objet de type point, la méthode retourne une liste contenant les coordonnées du point – ici Mo. `car` permet d'extraire le premier élément de cette liste, `cadr` le deuxième. Le reste du script ne devrait pas poser de problème.

Nommons ce script Y1 puis construisons le point M1 de coordonnées (X1 ; Y1) et enfin la tangente (MoM1).

Bien sûr, il aurait été possible de n'utiliser que deux ou trois gros scripts à la place de cette myriade de scripts. Mais nous espérons que ces petits exemples vous ont donné envie d'essayer par vous-même les DGS.

### 3.2.2 Méthodes de référence pour les scripts Dr. Geo

Les sections suivantes contiennent la description des méthodes disponibles pour les DGS. Elles sont classées par type d'objet géométrique ou numérique.

#### Point

**valeur** (`getAbscissa point`)

*point* : Référence d'un point libre sur une ligne

*Retourne* : L'abscisse de ce point sur la ligne. La valeur appartient à l'intervalle  $[0 ; 1]$

**Exemple:**

```
(define x (getAbscissa a1))
(* x 10)
```

**(setAbscissa point x)**

*point* : Référence d'un point libre sur une ligne

*x* : Valeur décimale de l'intervalle  $[0 ; 1]$  représentant la nouvelle abscisse

**Exemple:**

```
(setAbscissa a1 0.5)
```



`liste (getCoordinates point|vecteur)`

Retourne les coordonnées d'un point ou d'un vecteur.

*point/vecteur*: Référence vers un point ou un vecteur

*Retourne*: Liste contenant les coordonnées du point ou du vecteur

**Exemple:**

```
(define c (getCoordinates a1))
(define x (car c))
(define y (cadr c))
(+ (* x x) (* y y))
```

`(setCoordinates point coord)`

Positionne les coordonnées d'un point

*point*: Référence d'un point libre dans le plan

*coord*: Liste de deux nombres décimaux

**Exemple:**

```
(define l (list 1.4 (random 5)))
(setCoordinate a1 l)
```

### Droite, Demi-droite, Segment, Vecteur

`valeur (getSlope direction)`

*direction*: Référence vers un objet de type droite, demi-droite, segment ou vecteur

*Retourne*: La pente de cette direction

**Exemple:**

```
(define p (getSlope a1))
```

`valeur (getUnit direction)`

*direction*: Référence vers un objet de type droite, demi-droite, segment ou vecteur

*Retourne*: Une liste contenant les coordonnées d'un vecteur unitaire

**Exemple:**

```
(define v (getUnit a1))
```

`valeur (getNormal direction)`

*direction*: Référence vers un objet de type droite, demi-droite, segment ou vecteur

*Retourne*: Une liste contenant les coordonnées d'un vecteur normal à la direction

**Exemple:**

```
(define n (getNormal a1))
```

`valeur (getNorm vecteur)`

*vecteur*: Référence vers un vecteur

*Retourne*: La norme de ce vecteur

**Exemple:**

```
(define n (getNorm a1))
```

`valeur (getLength segment)`

*segment*: Référence vers un segment

*Retourne*: La longueur de ce segment





**Exemple:**

```
(define l (getLength a1))
```

**Cercle, Arc de cercle**

```
liste (getCenter cercle|arc-cercle)
```

*cercle/arc-cercle*: Référence vers un cercle ou un arc de cercle

*Retourne*: Liste contenant les coordonnées du centre du cercle ou de l'arc de cercle

**Exemple:**

```
(define c (getCenter a1))
(car c)
```

```
valeur (getRadius cercle|arc-cercle)
```

*cercle/arc-cercle*: Référence vers un cercle ou un arc de cercle

*Retourne*: Rayon du cercle ou de l'arc de cercle

**Exemple:**

```
(define r (getRadius a1))
```

```
valeur (getLength cercle|arc-cercle)
```

*cercle/arc-cercle*: Référence vers un cercle ou un arc de cercle

*Retourne*: Périmètre du cercle ou longueur de l'arc de cercle

**Exemple:**

```
(define l (getLength a1))
```

**Nombre**

```
valeur (getValue nombre)
```

*nombre*: Référence vers un nombre

*Retourne*: Valeur de ce nombre

**Exemple:**

```
(define a (getValue a1))
(define b (getValue a2))
(+ a b)
```

```
(setValue nombre v)
```

*nombre*: Référence vers un nombre

*v*: Valeur décimale

**Exemple:**

```
(define v (getValue a1))
(setValue a2 v)
```

**Angle**

```
valeur (getAngle angle)
```

*angle*: Référence vers un angle, orienté ou géométrique

*Retourne*: Une mesure de cet angle en degrés. Pour obtenir une mesure en radians, utiliser



la méthode `getValue`

**Exemple:**

```
(define angle1 (getAngle a1))
(define angle2 (getAngle2))
(define angle3 (getAngle a3))
(+ angle1 angle2 angle3)
```

**Autres**

```
(move item t)
```

*item*: Référence vers un objet de la figure

*t*: Vecteur à deux dimensions

**Exemple:**

```
(define v (vector .1 0))
(move a1 v)
```

### 3.3 Figure Scheme de Dr. Geo

Les *Figures Scheme de DR. GEO* – (FSD) – sont des figures écrites dans un langage relativement naturel. Il ne s'agit donc plus de construire une figure à l'aide de l'interface graphique de DR. GEO mais plutôt de décrire une figure dans le langage Scheme. Nous avons apporté le plus grand soin afin que la syntaxe utilisée soit facile et légère. Aussi l'ensemble des mots clés utilisés pour décrire une figure simple sont adaptables dans différentes langues. Ainsi une figure pourra être décrite en Français, en Anglais, en Espagnol, etc. (Un mélange de langues est même possible mais ce n'est pas souhaitable).

#### 3.3.1 Quelques exemples

En lui-même Scheme est un langage de très haut niveau, lorsqu'une figure est définie dans ce langage, nous disposons également de toute sa puissance pour par exemple définir récursivement telle partie de la figure, ou bien pour placer aléatoirement certains objets de telle sorte qu'à chaque ouverture de la figure, celle-ci est légèrement différente. Bref, les FSD sont libérées du carcan de l'interface graphique tout en étant renforcées du langage Scheme. Une FSD est donc un fichier d'extension **.scm** créé à l'aide d'un éditeur de texte, il est ensuite ouvert dans DR. GEO à l'aide de la commande **Fichier->Évaluer**.

Commençons par étudier un exemple simple de FSD :

```
(nouvelle-figure "Ma figure")
```

C'est la plus petite FSD que nous puissions définir. Lors de son chargement dans DR. GEO, celle-ci va simplement créer une nouvelle figure vide nommée "Ma figure". Nous pourrions multiplier les commandes **(nouvelle-figure "Ma figure")** autant de fois que nous le souhaitons, autant de figures seraient créées.

Abordons un deuxième exemple :

```
(nouvelle-figure "Ma figure")
(soit Point "A" libre 1.2 -2)
```

Cette FSD définit une figure avec un point libre *A* de coordonnées initiales (1,2 ; -2). Comme nous pouvons le voir la syntaxe de définition d'un objet géométrique est relativement agréable, d'autant plus qu'elle est exprimée dans une langue naturelle. Intéressons-nous de plus près à la deuxième instruction, en effet celle-ci suit une syntaxe qui est commune à toutes les commandes de définition d'objet. Ce type de commande se décompose comme suit :



1. Elle commence toujours par le mot-clé **soit**, il indique que nous souhaitons définir un nouvel objet.
2. Il est immédiatement suivi de la catégorie de l'objet, ici **Point**.
3. Le nom de l'objet vient ensuite, *A*, il doit toujours être entouré de ". Si nous ne souhaitons pas nommer l'objet, il faut tout de même donner un nom vide comme suit : "".
4. Enfin, nous précisons le type de l'objet – le type de point dans notre exemple – ici **libre**. Cela signifie que le point *A* est libre.
5. Le type de l'objet est suivi d'une liste d'arguments spécifiques. Dans notre exemple cette liste est composée de deux nombres, les coordonnées du point libre *A*.

Poursuivons avec un troisième exemple :

```
(define (triangle p1 p2 p3)
  (Segment "" extrémités p1 p2)
  (Segment "" extrémités p2 p3)
  (Segment "" extrémités p1 p3))

(define (hasard)
  (- 8 (* 16 (random:uniform))))

(nouvelle-figure "Ma figure")

(soit Point "A" libre (hasard) 0)
(soit Point "B" libre 5 0)
(soit Point "C" libre (hasard) 5)

(triangle A B C)
```

Cet exemple est particulièrement intéressant, il nous montre trois choses importantes :

1. L'introduction de construction de plus haut niveau, non prévue au départ par DR. GEO. Ici nous avons défini la fonction **triangle** qui, à partir de trois points, construit le triangle passant par ces trois points. Nous pouvons comparer ceci avec les macro-constructions mais avec un degré de liberté beaucoup plus important.
2. La définition de fonctions associées, ici nous avons défini la fonction **hasard** qui retourne un nombre décimal compris entre -8 et 8. Nous utilisons cette fonction pour placer au hasard certains points de notre figure, ainsi à chaque ouverture la figure est légèrement différente.
3. En fait l'utilisation du mot clé **soit** n'est pas obligatoire, nous l'utilisons lorsque nous souhaitons garder une référence de l'objet créé. Par exemple dans la fonction **triangle**, nous ne gardons pas de références des segments créés, en revanche lorsque nous définissons nos points *A*, *B* et *C* nous avons besoin de garder une référence, ces références ont le même nom<sup>8</sup> sans guillemet : *A*, *B* et *C*. Dans la suite nous appellerons **symbole** ces références, c'est la terminologie exacte du langage Scheme. Ainsi, lors de l'appel de la fonction **triangle**, nous passons en paramètre les symboles *A*, *B* et *C* qui sont utilisés pour définir nos trois segments.

Noter que lors de la définition des segments, nous ne donnons pas de nom, dans ce cas DR. GEO va attribuer un nom par défaut défini à partir du nom des extrémités. Nos trois segments auront donc comme nom  $[AB]$ ,  $[BC]$  et  $[AC]$ .

Pour clore cette section, voici un dernier exemple :

<sup>8</sup>D'un point de vue interne au langage Scheme, ces références sont des symboles pointant vers une structure interne à l'objet – un prototype – alors que les noms sont des chaînes de caractères.



```
(soit Point "A" libre 1 0)
(soit Point "B" libre 5 0)
(soit Droite "d1" 2points A B)

(envoi A couleur jaune)
(envoi A forme rond)
(envoi A taille large)
(envoi B masquer)
(envoi d1 épaisseur tiret)
```

Les trois premières commandes créent deux points et une droite. La partie qui nous intéresse plus particulièrement est la commande **envoi**. Cette commande permet de communiquer avec un objet dont nous avons gardé un symbole, ici nous avons les symboles **A**, **B** et **d1**. Elle consiste à envoyer un message à un objet, son premier argument est l'objet avec lequel nous communiquons, le deuxième argument le message, le troisième et les suivants sont déterminés par la nature du message. Par exemple **(envoi A couleur jaune)** envoie le message **couleur** avec comme paramètre **jaune**, le point **A** est peint en jaune. Il est assez facile de comprendre le sens des autres commandes **envoi**. Elles seront expliquées dans la section suivante.

Nous avons terminé notre petite visite guidée des *Figure Scheme Dr. Geo*. Dans les sections suivantes nous exposons l'ensemble des commandes disponibles pour définir des FSD.

### 3.3.2 Méthodes de référence pour les Figures Scheme Dr. Geo

La définition d'objets dans un document FSD se fait par l'intermédiaire de prototypes. Les prototypes sont en quelque sorte des objets qu'il est possible d'interroger et de modifier comme nous le verrons par la suite.

Cependant, avant toute définition d'objets d'une figure, cette dernière doit être créée avec la commande **nouvelle-figure**.

#### Commandes générales

```
(nouvelle-figure nom)
```

*nom*: Chaîne de caractères

*Retourne*: Ne retourne pas de valeur. L'appel ne produit qu'un effet de bord, à savoir la création d'une nouvelle figure, les objets suivants sont créés dans cette figure, jusqu'au prochain appel de ce type.

#### Exemple:

```
(nouvelle-figure "Ma 1er figure")
```

#### Définition d'objets d'une figure

Un objet peut être défini par l'intermédiaire de différentes syntaxes :

- **(soit Point "p1" type args)**  
Le point est créé et sa référence est sauvée dans la variable **p1**. Cette syntaxe utilise une macro Scheme.
- **(Point "Nom" type args)**  
Le point est créé mais aucune référence du point n'est conservée.
- **(define p1 (Point "Nom" type args))**  
Le point est créé et sa référence est sauvée dans la variable **p1**.
- **(set! p1 (Point "Nom" type args))**  
Le point est créé et sa référence est copiée dans la variable préexistante **p1**.



Si des objets sont créés depuis le corps d'une fonction, utiliser soit la forme **set !** ou la forme spéciale Scheme **let**. Il est important de remarquer que l'appel de base est celui d'une fonction retournant une référence de l'objet créé.

Pour en savoir plus sur la correspondance entre les noms de commande Scheme en français et en anglais, voir le fichier `/usr/share/drgeo/scm/drgeo_scm_interface_constant_fr.scm`.

#### Point

```
prototype (Point nom libre x y)
```

*nom*: Chaîne de caractères désignant le nom de l'objet

*x*: L'abscisse du point

*y*: L'ordonnée du point

*Retourne*: Référence d'un point libre du plan de coordonnées initiales *x* et *y*.

#### Exemple:

```
(define p1 (Point "A" libre 1.2 (acos -1)))
```

```
prototype (Point nom sur-courbe ligne x)
```

*nom*: Chaîne de caractères désignant le nom de l'objet

*ligne*: Référence d'une ligne (droite, demi-droite, segment, etc.)

*x*: Abscisse curviligne du point libre, la valeur appartient à l'intervalle  $[0; 1]$

*Retourne*: Référence d'un point libre sur une ligne.

#### Exemple:

```
(Point "M" sur-courbe s1 0.5)
```

```
prototype (Point nom milieu-2pts p1 p2)
```

*nom*: Chaîne de caractères désignant le nom de l'objet

*p1*: Référence d'un point

*p2*: Référence d'un point

*Retourne*: Référence du milieu des deux points.

#### Exemple:

```
(soit Point "A" libre 1 1)
```

```
(soit Point "B" libre 4 4)
```

```
(Point "I" milieu-2pts A B)
```

```
prototype (Point nom milieu-segment s)
```

*nom*: Chaîne de caractères désignant le nom de l'objet

*s*: Référence d'un segment

*Retourne*: Référence du milieu du segment.

#### Exemple:

```
(Point "L" milieu-segment s)
```

```
prototype (Point nom intersection l1 l2)
```

*nom*: Chaîne de caractères désignant le nom de l'objet

*l1*: Référence d'une ligne

*l2*: Référence d'une ligne

*Retourne*: Référence du point d'intersection des deux lignes.

#### Exemple:

```
(Point "I" intersection droite segment)
```



**prototype** (Point nom intersection2 l1 l2)

*nom*: Chaîne de caractères désignant le nom de l'objet

*l1*: Référence d'une ligne

*l2*: Référence d'une ligne

*Retourne*: Référence du 2ème point d'intersection des deux lignes lorsqu'une des deux est du type arc de cercle ou cercle.

**Exemple:**

(Point "I" intersection2 droite cercle)

### Droite

**prototype** (Droite nom 2points p1 p2)

*nom*: Chaîne de caractères désignant le nom de l'objet

*l1*: Référence d'un point

*l2*: Référence d'un point

*Retourne*: Référence d'une droite passant par deux points.

**Exemple:**

(soit Point "A" libre 0 0)

(soit Point "M" libre 1 2)

(Droite "" 2points A M)

**prototype** (Droite nom parallèle p d)

*nom*: Chaîne de caractères désignant le nom de l'objet

*p*: Référence d'un point

*d*: Référence d'une direction (droite, segment, vecteur, ...)

*Retourne*: Référence d'une droite parallèle à la direction de *d* et passant par *p*.

**Exemple:**

(soit Point "A" libre 1 5)

(soit Droite "d1" parallèle A d)

**prototype** (Droite nom perpendiculaire p d)

*nom*: Chaîne de caractères désignant le nom de l'objet

*p*: Référence d'un point

*d*: Référence d'une direction (droite, segment, vecteur, ...)

*Retourne*: Référence d'une droite perpendiculaire à la direction de *d* et passant par *p*.

**Exemple:**

(soit Point "A" libre 1 5)

(soit Droite "d1" perpendiculaire A d)

### Demi-droite

**prototype** (Demi-droite nom 2points o p)

*nom*: Chaîne de caractères désignant le nom de l'objet

*o*: Référence d'un point, origine de la demi-droite

*p*: Référence d'un point, point de la demi-droite

*Retourne*: Référence d'une demi-droite définie par son origine et un point.

**Exemple:**

(soit Point "A" libre 1 5)

(soit Point "O" libre 0 0)

(soit Demi-droite "dd1" 2points A O)



**Segment**

```
prototype (Segment nom extrémités p1 p2)
```

*nom*: Chaîne de caractères désignant le nom de l'objet

*p1*: Référence d'un point

*p2*: Référence d'un point

*Retourne*: Référence d'un segment défini par ses extrémités.

**Exemple:**

```
(soit Point "A" libre 1 5)
```

```
(soit Point "B" libre 10 4)
```

```
(soit Segment "" extrémités A B)
```

**Cercle**

```
prototype (Cercle nom 2points c p)
```

*nom*: Chaîne de caractères désignant le nom de l'objet

*c*: Référence d'un point, centre du cercle

*p*: Référence d'un point sur le cercle

*Retourne*: Référence d'un cercle défini par son centre et un point.

**Exemple:**

```
(soit Point "A" libre 1 5)
```

```
(soit Point "B" libre 10 4)
```

```
(soit Cercle "C1" 2points A B)
```

```
prototype (Cercle nom centre-rayon c r)
```

*nom*: Chaîne de caractères désignant le nom de l'objet

*c*: Référence d'un point, centre du cercle

*r*: Référence d'une valeur numérique, rayon du cercle

*Retourne*: Référence d'un cercle défini par son centre et son rayon.

**Exemple:**

```
(soit Point "A" libre 1 5)
```

```
(soit Nombre "r" libre 10)
```

```
(soit Cercle "C1" centre-rayon A r)
```

```
prototype (Cercle nom centre-segment c s)
```

*nom*: Chaîne de caractères désignant le nom de l'objet

*c*: Référence d'un point, centre du cercle

*s*: Référence d'un segment dont la longueur est le rayon du cercle

*Retourne*: Référence d'un cercle défini par son centre et son rayon.

**Exemple:**

```
(soit Point "A" libre 1 5)
```

```
(soit Cercle "C1" centre-rayon A s)
```

**Arc de cercle**

```
prototype (Arc-cercle nom 3points p1 p2 p3)
```

*nom*: Chaîne de caractères désignant le nom de l'objet

*p1*: Référence d'un point, 1<sup>ère</sup> extrémité de l'arc

*p2*: Référence d'un point de l'arc



*p3*: Référence d'un point, 2<sup>ème</sup> extrémité de l'arc

*Retourne*: Référence d'un arc de cercle défini par ses extrémités et un point.

**Exemple:**

```
(soit Point "A" libre 1 5)
(soit Point "B" libre 0 5)
(soit Point "C" libre -1 -2)
(soit Arc-cercle "arc" 3points A B C)
```

### Polygone

**prototype (Polygone nom npoints args)**

*nom*: Chaîne de caractères désignant le nom de l'objet

*args*: Une liste de références de points; sommets du polygone

*Retourne*: Référence d'un polygone défini par ses sommets.

**Exemple:**

```
(soit Polygone "quad" npoints A B C D)
```

### Les transformations géométriques

Les prototypes des transformations géométriques permettent la construction des transformés d'objets. Elles s'appliquent à des références d'objets de type point, segment, droite, demi-droite, vecteur, cercle, arc de cercle et polygone.

**prototype (TypeObjet nom rotation objet centre angle)**

*TypeObjet*: Point, Segment, Droite, Demi-droite, Vecteur, Cercle, Arc-cercle, Polygone

*nom*: Chaîne de caractères désignant le nom de l'objet

*objet*: Référence de l'objet à transformer

*centre*: Référence d'un point, centre de la rotation

*angle*: Référence d'une valeur, angle de la rotation

*Retourne*: Référence de l'objet transformé.

**Exemple:**

```
(soit Point "I1" rotation I C a)
```

**prototype (TypeObjet nom homothétie objet centre k)**

*TypeObjet*: Point, Segment, Droite, Demi-droite, Vecteur, Cercle, Arc-cercle, Polygone

*nom*: Chaîne de caractères désignant le nom de l'objet

*objet*: Référence de l'objet à transformer

*centre*: Référence d'un point, centre de l'homothétie

*k*: Référence d'une valeur, facteur de l'homothétie

*Retourne*: Référence de l'objet transformé.

**Exemple:**

```
(soit Polygone "P1" homothétie P C k1)
```

**prototype (TypeObjet nom symétrie objet centre)**

*TypeObjet*: Point, Segment, Droite, Demi-droite, Vecteur, Cercle, Arc-cercle, Polygone

*nom*: Chaîne de caractères désignant le nom de l'objet

*objet*: Référence de l'objet à transformer

*centre*: Référence d'un point, centre de la symétrie

*Retourne*: Référence de l'objet transformé.

**Exemple:**

```
(soit Segment "S1" symétrie S C)
```





```
prototype (TypeObjet nom reflexion objet axe)
```

*TypeObjet*: Point, Segment, Droite, Demi-droite, Vecteur, Cercle, Arc-cercle, Polygone

*nom*: Chaîne de caractères désignant le nom de l'objet

*objet*: Référence de l'objet à transformer

*axe*: Référence d'une droite, axe de la réflexion

*Retourne*: Référence de l'objet transformé.

**Exemple:**

(soit Polygone "P1" reflexion P d1)

```
prototype (TypeObjet nom translation objet vecteur)
```

*TypeObjet*: Point, Segment, Droite, Demi-droite, Vecteur, Cercle, Arc-cercle, Polygone

*nom*: Chaîne de caractères désignant le nom de l'objet

*objet*: Référence de l'objet à transformer

*vecteur*: Référence d'un vecteur

*Retourne*: Référence de l'objet transformé.

**Exemple:**

(soit Cercle "C1" translation C v)

### Lieu géométrique

```
prototype (Lieu nom 2points m c)
```

*nom*: Chaîne de caractères désignant le nom de l'objet

*m*: Référence d'un point mobile sur une ligne

*c*: Référence d'un point fixe dépendant du point m

*Retourne*: Référence d'un lieu.

**Exemple:**

(Lieu "lieu1" 2points M I)

### Vecteur

```
prototype (Vecteur nom 2points o e)
```

*nom*: Chaîne de caractères désignant le nom de l'objet

*o*: Référence d'un point, origine du vecteur

*e*: Référence d'un point, extrémité du vecteur

*Retourne*: Référence d'un vecteur.

**Exemple:**

(soit Point "B" libre 0 5)

(soit Point "C" libre -1 -2)

(Vecteur "" 2points C B)

### Nombre

```
prototype (Nombre nom libre x y v)
```

*nom*: Chaîne de caractères désignant le nom de l'objet

*x, y*: Les coordonnées de l'emplacement du nombre

*v*: La valeur initiale du nombre

*Retourne*: Référence d'un nombre libre.

**Exemple:**

(soit Nombre "pi" libre 5 5 (acos -1))



**prototype** (Nombre nom longueur-segment x y s)

*nom*: Chaîne de caractères désignant le nom de l'objet

*x,y*: Les coordonnées de l'emplacement du nombre

*s*: Référence d'un segment

*Retourne*: Référence d'un nombre, longueur d'un segment.

**Exemple:**

(soit Nombre "l" longueur-segment 5 5 S)

**prototype** (Nombre nom norme-vecteur x y v)

*nom*: Chaîne de caractères désignant le nom de l'objet

*x,y*: Les coordonnées de l'emplacement du nombre

*s*: Référence d'un vecteur

*Retourne*: Référence d'un nombre, norme d'un vecteur.

**Exemple:**

(soit Nombre "l" norme-vecteur 5 5 V)

**prototype** (Nombre nom point-cercle x y p c)

*nom*: Chaîne de caractères désignant le nom de l'objet

*x,y*: Les coordonnées de l'emplacement du nombre

*p*: Référence d'un point

*c*: Référence d'un cercle

*Retourne*: Référence d'un nombre, distance entre le point et le cercle.

**Exemple:**

(soit Nombre "l" point-cercle 5 5 P C)

**prototype** (Nombre nom point-droite x y p d)

*nom*: Chaîne de caractères désignant le nom de l'objet

*x,y*: Les coordonnées de l'emplacement du nombre

*p*: Référence d'un point

*c*: Référence d'une droite

*Retourne*: Référence d'un nombre, distance entre le point et la droite.

**Exemple:**

(soit Nombre "d" point-droite 5 5 M D1)

**prototype** (Nombre nom point-point x y p1 p2)

*nom*: Chaîne de caractères désignant le nom de l'objet

*x,y*: Les coordonnées de l'emplacement du nombre

*p1*: Référence d'un point

*p2*: Référence d'un point

*Retourne*: Référence d'un nombre, distance entre les deux points.

**Exemple:**

(soit Nombre "d" point-point 5 5 A B)

**prototype** (Nombre nom longueur-cercle x y c)

*nom*: Chaîne de caractères désignant le nom de l'objet

*x,y*: Les coordonnées du nombre

*c*: Référence d'un cercle

*Retourne*: Référence d'un nombre, longueur d'un cercle.



**Exemple:**

(soit Nombre "p" longueur-cercle 5 5 C)

```
prototype (Nombre nom pente-droite x y d)
```

*nom*: Chaîne de caractères désignant le nom de l'objet

*x,y*: Les coordonnées du nombre

*d*: Référence d'une droite

*Retourne*: Référence d'un nombre, pente d'une droite.

**Exemple:**

(soit Nombre "p" pente-droite 5 5 d1)

```
prototype (Nombre nom longueur-arc x y arc)
```

*nom*: Chaîne de caractères désignant le nom de l'objet

*x,y*: Les coordonnées du nombre

*arc*: Référence d'un arc de cercle

*Retourne*: Référence d'un nombre, longueur d'un arc de cercle.

**Exemple:**

(soit Nombre "l" longueur-arc 5 5 ABC)

**Angle**

```
prototype (Angle nom géométrique A B C)
```

*nom*: Chaîne de caractères désignant le nom de l'objet

*A*: Référence d'un point

*B*: Référence d'un point, sommet de l'angle

*C*: Référence d'un point

*Retourne*: Référence d'un angle géométrique.

**Exemple:**

(soit Angle "a" géométrique A B C)

```
prototype (Angle nom orienté x y v1 v2)
```

*nom*: Chaîne de caractères désignant le nom de l'objet

*x,y*: Les coordonnées de l'emplacement de l'angle

*v1*: Référence d'un vecteur

*v2*: Référence d'un vecteur

*Retourne*: Référence d'un angle orienté formé par les deux vecteurs.

**Exemple:**

```
(define v1 (Vecteur "" 2points A B))
```

```
(define v2 (Vecteur "" 2points A C))
```

```
(Angle "a" orienté 1 1 v1 v2)
```

**Modification d'attributs d'objets**

Pour modifier les attributs d'un objet déjà créé, nous utilisons un système de messages envoyés directement au prototype représentant l'objet en question. La modification des attributs se fait donc toujours à posteriori.

```
(envoi objet couleur valeur)
```

*objet*: Référence d'un objet



*valeur*: La couleur, les valeurs possibles sont noir, gris-noir, gris, blanc, vert-noir, vert, bleu-noir, bleu, rouge, bordeaux, jaune, orange

**Exemple:**

(soit Point "A" libre 1 2)

(envoi A couleur vert)

(envoi ligne épaisseur valeur)

*ligne*: Référence d'une ligne (droite, demi-droite, cercle, lieu, etc.)

*valeur*: L'épaisseur, les valeurs possibles sont tiret, normal, large

**Exemple:**

(soit Point "A" libre 1 2)

(soit Point "O" libre 0 0)

(soit Droite "d" 2points A B)

(envoi d épaisseur tiret)

(envoi point taille valeur)

*point*: Référence d'un point

*valeur*: La taille du point, les valeurs possibles sont petit, normal, large

**Exemple:**

(soit Point "A" libre 1 2)

(envoi A taille petit)

(envoi point forme valeur)

*point*: Référence d'un point

*valeur*: La forme du point, les valeurs possibles sont rond, croix, rond-vide, rec, rec-vide

**Exemple:**

(soit Point "A" libre 1 2)

(envoi A forme rond)

(envoi objet masquer)

*objet*: Référence d'un objet à masquer

**Exemple:**

(soit Point "A" libre 1 2)

(envoi A masquer)

### 3.3.3 Synonymes des commandes Figure Scheme Dr. Geo

Cette section consiste en une transcription légèrement modifiée du fichier (/usr/share/drgeo/scm/drgeo\_scm\_interface\_constant\_fr.scm).

Ce fichier définit les synonymes des commandes utilisées pour écrire une Figure Scheme en français. Ces synonymes sont toujours définis à partir de la version de référence des commandes qui est en anglais.

Cette section peut donc vous aider à comprendre une Figure Scheme écrite en anglais.

En fait il est possible de combiner des commandes en anglais avec leurs synonymes en français, espagnol, etc. Cependant, si vous souhaitez que votre figure soit exploitable par n'importe quel groupe linguistique, nous vous suggérons d'utiliser l'anglais.

Par exemple, les portions de codes suivantes sont synonymes :



```
(lets Point "P" free 3 3)(send A 'color bordeaux)
(soit Point "P" libre 3 3)(envoi A couleur bordeaux)
```

Certains termes anglais n'ont pas de synonymes en français. C'est tout simplement parce que leur version française est identique.

## COULEURS :

```
noir black
gris-noir dark-grey
gris grey
blanc white
vert-noir dark-green
vert green
bleu-noir dark-blue
bleu blue
rouge red
jaune yellow
```

## ÉPAISSEURS :

```
tiret dashed
petit small
```

## FORMES DE POINT :

```
rond round
croix cross
rond-vide round-empty
rec-vide rec-empty
```

## STYLES :

```
couleur 'color
épaisseur 'thickness
forme 'shape
taille 'size
masquer 'masked
```

## POINTS :

```
libre 'free
sur-courbe 'on-curve
milieu-2pts 'middle-2pts
milieu-segment 'middle-segment
```

## SEGMENTS :

```
extrémités 'extremities
```

## DROITES :

```
parallèle 'parallel
```

```
orthogonale 'orthogonal
```

```
perpendiculaire 'orthogonal
```

## CERCLES :

```
centre-rayon 'center-radius
```

```
centre-segment 'center-segment
```

## NUMÉRIQUES :

```
longueur-segment 'segment-length
```

```
norme-vecteur 'vector-norm
```

```
point-droite 'point-line
```

```
point-cercle 'point-circle
```

```
longueur-cercle 'circle-length
```

```
pente-droite 'line-slope
```

```
longueur-arc 'arc-length
```

## ANGLES :

```
géométrique 'geometric
```

```
orienté 'oriented
```

## TRANSFORMATIONS :

```
homothétie 'scale
```

```
symétrie 'symmetry
```

## GÉNÉRAL :

```
soit lets
```

```
nouvelle-figure new-figure
```

```
envoi send
```

```
Droite Line
```

```
Demi-droite Ray
```

```
Cercle Circle
```

```
Arc-cercle Arc
```

```
Lieu Locus
```

```
Vecteur Vector
```

```
Nombre Numeric
```

```
Polygone Polygon
```

### 3.3.4 Galerie d'exemples

Pour illustrer l'utilisation des Figures Scheme DR. GEO, nous vous proposons une petite série d'exemples. Ceux-ci vous montrent leurs importantes possibilités et nous espérons qu'ils seront également une source d'inspiration. Pour chacun de ces exemples, nous donnons le code source Scheme de la figure puis son résultat. Le code source peut être copié dans un éditeur de texte puis sauvegardé dans un fichier d'extension **.scm** pour l'évaluer avec DR. GEO.



### Polygone régulier

Construire un polygone régulier, avec un nombre de cotés arbitraire, peut se faire par l'intermédiaire d'une fonction récursive Scheme.

```
(define pi (acos -1))
(define n 15)
(define x0 0)
(define y0 0)
(define p1 0)

(define (polygon center p a n)
  (if (> n 0)
      (begin
        (set! p1 (Point "" rotation p center a))
        (send p1 masked)
        (Segment "" extremities p p1)
        (polygon center p1 a (- n 1))))))

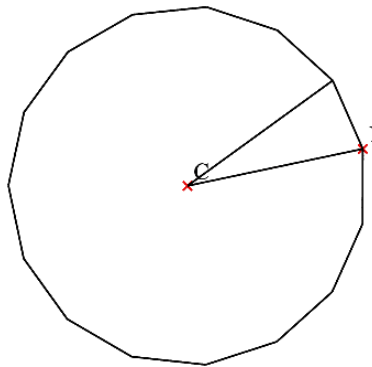
(new-figure "Regular Polygon!")
(lets Point "C" free x0 y0)
(lets Numeric "a" free 0 0 (* 2 (/ pi n)))
(send a masked)

(set! p1 (Point "I" free 5 0))

(lets Segment "S" extremities C p1)
(Segment "" rotation S C a)

(polygon C p1 a n)
```

FIG. 3.12 – Un polygone régulier à 15 côtés



### Fractal

La construction d'une courbe fractale de la forme d'un arbre se fait très aisément avec une figure Scheme. Le code source de la figure est étonnement compact, surtout comparé à une construction "à la main" depuis l'interface graphique.



```

(new-figure "Baum")
(lets Numeric "A1" free 2 2 +3.4)
(lets Numeric "A2" free 2 3 -3.7)
(lets Numeric "S1" free 2 4 +0.5)
(lets Numeric "S2" free 2 5 +0.9)

(define (dec n)
  (- n 1))
(define (inc n)
  (+ n 1))
(define (invisible p)
  (send p masked)
  p)
(define (scalerot oP C a s)
  (let* ((sP (invisible (Point "" scale oP C s)))
        (rP (invisible (Point "" rotation sP C a))) )
    rP))
(define (Zweig p0 p1 n)
  (Segment "" extremities p0 p1)
  (let* ((left-scale (if (odd? n) S1 S2))
        (left-angle A1)
        (right-scale (if (odd? n) S2 S1))
        (right-angle A2) )
    (if (> n 0)
      (begin
        (Zweig p1 (scalerot p0 p1 left-angle left-scale) (dec n))
        (Zweig p1 (scalerot p0 p1 right-angle right-scale) (dec n))))))

(lets Point "A" free -3 0)
(lets Point "B" free -3 2)
(Zweig A B 6)

```

## 3.4 Masquer des outils dans l'interface

DR. GEO offre la possibilité de préparer des sessions<sup>9</sup> dans lesquelles l'enseignant peut décider, pour certaines figures, de bloquer l'accès à certains outils. Le verrouillage se fait à la figure près avec un mot de passe différent à chaque fois. Cela donne toute la latitude à l'enseignant de préparer une activité avec différentes figures et autant de degrés de verrouillage.

### 3.4.1 Verrouiller des outils

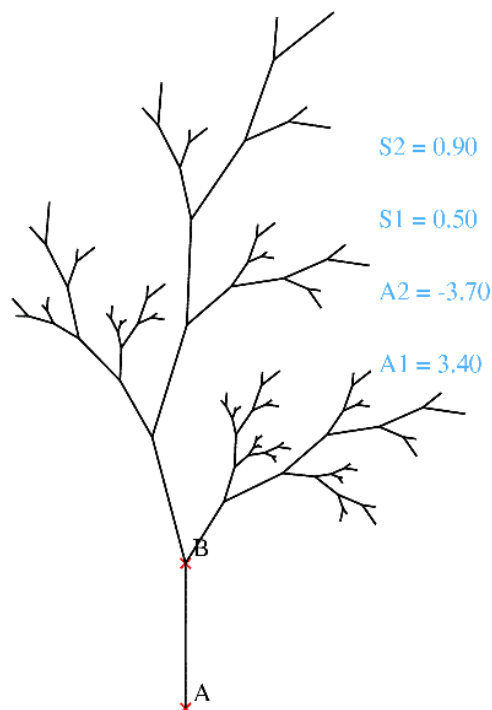
Le verrouillage se fait à la figure près, aussi depuis l'une d'entre elles, aller au menu **Édition->Adapter l'Interface**. Une grande boîte de dialogue s'affiche, dans celle-ci l'utilisateur reconnaît les icônes utilisées dans la barre d'icônes de DR. GEO. Il peut en cliquant sur ces icônes, les désactiver ou activer. Lorsque un outil est désactivé, son icône représentative apparaît légèrement grisée. Il est également possible de désactiver toute une rangée d'outils en cliquant sur l'icône de haut niveau (i.e. celle avec un petit triangle vert).

Lorsque l'utilisateur a fini sa sélection d'outils à verrouiller, il procède au verrouillage à proprement parler en cliquant sur le bouton **Verrouiller** dans la boîte de dialogue, DR. GEO demande alors un mot de passe.

<sup>9</sup>Fichiers comprenant plusieurs documents DR. GEO (i.e. figure, macro, texte)



FIG. 3.13 – Une courbe fractale simulant la représentation d'un arbre




---

(!) Lors de la sauvegarde d'une figure ou d'une session avec des verrouillages de l'interface, les mots de passe sont également sauvegardés dans le fichier sous une forme cryptée.

---

### 3.4.2 Déverrouiller des outils

Il est bien sûr possible de déverrouiller une interface, que cela soit pour donner progressivement accès à des outils – lors d'un travail d'élèves sur une session – ou pour retravailler une figure. Pour cela, il faut aller dans le menu **Édition->Adapter l'Interface** et choisir dans la boîte de dialogue le bouton **Déverrouiller**. DR. GEO demande alors le mot de passe entré précédemment.

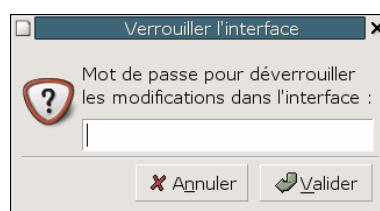




FIG. 3.14 – La boîte dialogue pour verrouiller des outils.



FIG. 3.15 – La boîte de dialogue pour déverrouiller l'interface d'une figure.





## Chapitre 4

# Fichiers et documents

Les constructions peuvent être enregistrées de deux manières. Une construction par fichier ou un ensemble de constructions par fichier (i.e. une session DR. GEO). Nous vous rappelons que les documents contenant des figures doivent être sauvegardés avec une extension **.fgeo** et ceux contenant uniquement des macro-constructions avec l'extension **.mgeo**. Pour les documents contenant des figures, macro-constructions et des textes d'explications, utiliser l'extension **.fgeo**.

Ceci n'est qu'indicatif mais suivre ces règles permet de s'y retrouver plus facilement parmi les fichiers.

### 4.1 Enregistrement d'une construction

À partir du menu **Fichier->Enregistrer** ou **Fichier->Enregistrer sous...**, un fichier contenant la figure de la vue active peut être enregistré.

---

(!) DR. GEO peut travailler avec plusieurs figures en même temps. L'utilisateur peut passer d'une figure à l'autre en cliquant sur l'onglet correspondant.

---

Avec le second menu, l'utilisateur peut changer le nom du document enregistré.

---

(!) Le nom de fichier par défaut proposé peut être changé à partir du menu **Édition->Préférences...** Pour plus d'information, voir la section comportement par défaut Section 2.3.1, page 21.

---

### 4.2 Enregistrement d'une session

Une session est un ensemble de données de DR. GEO que l'utilisateur veut enregistrer d'un seul coup dans un fichier. Cela permet à l'enseignant d'organiser un ensemble de données (figures, macro-constructions, notes) dans un seul fichier, de façon à faciliter leur exploitation.

À partir du menu **Fichier->Enregistrement multiple**, l'utilisateur peut ouvrir la boîte de dialogue de session.

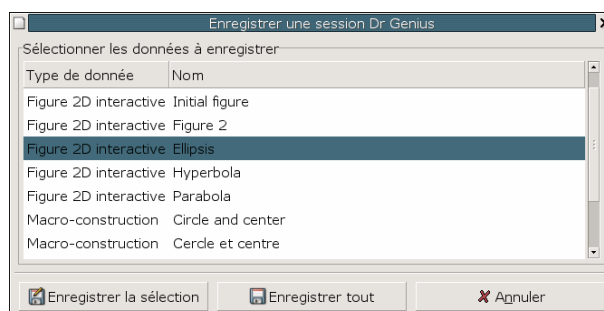
Dans cette boîte de dialogue, la liste de toutes les données actives est présentée dans un tableau. La première colonne liste les types des données contenues dans DR. GEO, la seconde les noms des données.

---

(!) Actuellement, une session peut contenir trois types de données : figure interactive 2D, macro-construction et texte.

---

FIG. 4.1 – La boîte dialogue de session DR. GEO



L'utilisateur peut décider de sélectionner une par une les données à enregistrer dans la liste puis appuyer sur le bouton **Enregistrer la sélection**. Autrement, il peut enregistrer toutes les données en appuyant sur le bouton **Enregistrer tout**.

---

(!) Le menu **Fichier->Enregistrement multiple** est le seul moyen d'enregistrer une macro-construction dans un fichier.

---

### 4.3 Enregistrer une macro-construction

Pour enregistrer une ou des macro-constructions dans un fichier, il faut procéder comme pour enregistrer une session – enregistrement multiple. Depuis la boîte de dialogue de sauvegarde d'une session, sélectionner la ou les macro-constructions à sauvegarder, puis sauver dans un fichier d'extension **.mgeo**. C'est tout !

Il est ainsi possible de vous composer des bibliothèques de macro-constructions, une par fichier ou plusieurs regroupées selon un même thème dans un seul fichier.

### 4.4 Ouvrir un fichier

Que l'utilisateur ait sauvegardé une seule figure ou une session avec différents types de documents, la procédure pour l'ouverture est la même par le menu **Fichier->Ouvrir**. Si la session ouverte contient des macro-constructions, celles-ci sont directement disponibles depuis l'outil permettant l'exécution des macro-constructions. Les macro-constructions sont disponibles depuis toutes les figures ouvertes.

### 4.5 Exporter une figure

DR. GEO offre la possibilité d'exporter une figure géométrique vers un document **L<sup>A</sup>T<sub>E</sub>X** ou PostScript. Ces deux formats d'exportation sont de type vectoriel contrairement à des images en plan de bits qui sont de qualité moindre pour une impression. Les commandes d'exportation sont accessibles depuis le sous-menu **Figure->Exporter sous...**

#### 4.5.1 Exportation L<sup>A</sup>T<sub>E</sub>X

Dans le cas d'une exportation vers **L<sup>A</sup>T<sub>E</sub>X**, le document exporté nécessite le paquet **pstricks**. Celui-ci est en général distribué avec **L<sup>A</sup>T<sub>E</sub>X**. Typiquement un document exporté en **L<sup>A</sup>T<sub>E</sub>X** pourra être intégré dans un autre document **L<sup>A</sup>T<sub>E</sub>X** ou même directement compilé :

```
latex figure.tex
dvips figure.dvi
```



Cette suite de commandes permet d'obtenir le document figure.ps qui peut être ouvert avec le logiciel GhostView<sup>1</sup> gv.

### 4.5.2 Exportation PostScript

L'exportation au format PostScript – extension **eps** pour *Encapsulated PostScript* – offre l'avantage d'être plus facilement utilisable depuis divers logiciels<sup>2</sup>. En fait le format EPS est plus ou moins un standard de fait en ce qui concerne les images vectorielles. Pour visualiser rapidement ce type d'image, nous utilisons le logiciel GhostView, sa commande est **gv**.

### 4.5.3 Exportation image png

Certains logiciels comme OpenOffice.org n'exploitent que partiellement le format EPS<sup>3</sup>. Aussi une exportation au format image PNG est également possible. L'image exportée sous ce format est en haute définition avec le fond de la figure encodé en zone transparente. Depuis un traitement de texte, cela peut permettre de faire un détournage de texte autour de la figure.

### 4.5.4 Exportation Fly Draw

Le serveur d'exercices WIMS développé par Dr. Xiao Gang à l'université de Nice dispose d'un format de description des figures appelé Fly Draw. DR. GEO propose également une exportation vers ce format.

### 4.5.5 Définir la zone exporter

Par défaut, DR. GEO exporte la zone visible de la figure. Ainsi si nous souhaitons exporter une zone précise de la figure, nous pouvons redimensionner la fenêtre de DR. GEO jusqu'à obtenir la zone voulue<sup>4</sup>.

Cependant cette solution, si elle est rapide et simple, n'est pas toujours suffisamment souple, de plus la zone d'exportation n'est pas enregistrée avec la figure. Une autre façon est donc d'utiliser la commande **Définir la zone d'exportation** accessible depuis le menu **Fichier->Préférences d'exportation**.

La commande activée, nous pouvons définir dans la figure un rectangle correspondant à la zone exportée. Cette zone peut être redéfinie autant de fois que nécessaire. Elle est représentée par un rectangle gris clair. Lorsque la figure est sauvegardée cette zone l'est également. Enfin elle peut être supprimée par la commande **Supprimer la zone d'exportation**.

<sup>1</sup>GhostView est un logiciel qui permet de visualiser des documents PostScript (.ps ou .eps) ainsi que des documents PDF

<sup>2</sup>Ce format est reconnu par TeXmacs, OpenOffice.org, Lyx, L<sup>A</sup>T<sub>E</sub>X Xfig, The Gimp et bien d'autres encore

<sup>3</sup>OpenOffice.org permet d'intégrer une image EPS mais pas de l'afficher. C'est uniquement à l'impression que celle-ci apparaît.

<sup>4</sup>Éventuellement, le panneau latéral de l'arbre logique de la figure peut être déplié pour réduire davantage la zone d'exportation.



FIG. 4.2 – Une figure avec une zone d’exportation définie

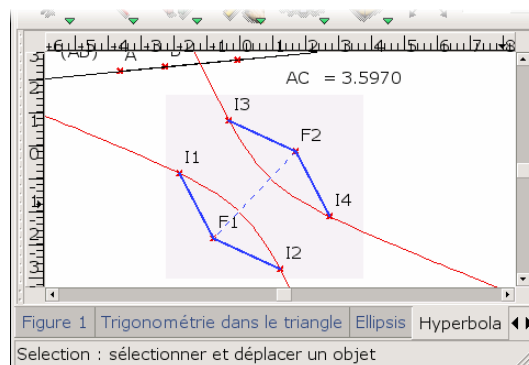
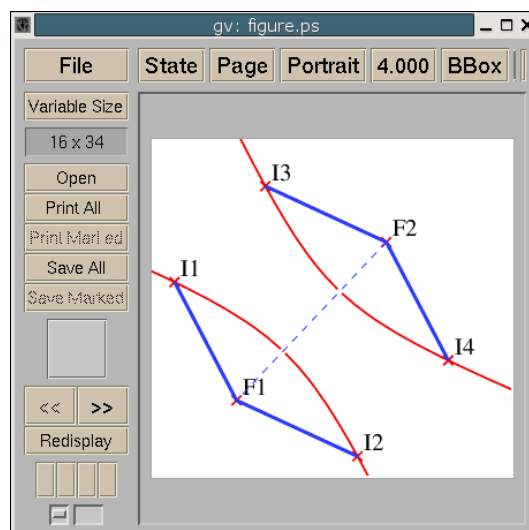


FIG. 4.3 – La zone d’exportation a été exportée dans un document PostScript, il est visualisé avec GhostView



## Chapitre 5

# Applications didactiques

Ce chapitre est une aide pour l'utilisateur souhaitant étudier DR. GEO à partir d'exemples. Contrairement aux chapitres précédents, l'approche est plus concrète par rapport à des situations précises. Le contenu de ce chapitre s'est constitué lors de diverses activités pédagogiques.

### 5.1 Pythagore et scripts

Une des possibilités d'utilisation didactique de DR. GEO consiste dans l'utilisation des scripts Scheme Section 3.2, page 28 pour résoudre des exercices de géométrie.

Comme exemple, nous allons montrer la solution d'un problème classique mettant en oeuvre le théorème de Pythagore dont le texte est le suivant :

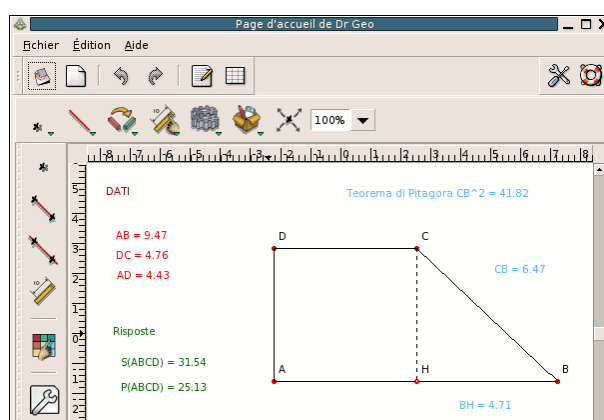
*Soit un trapèze rectangle  $ABCD$  où sont connues les bases et la hauteur. Calculer le périmètre et l'aire du trapèze.*

Il n'est pas difficile, si vous suivez le même modèle, de développer d'autres exemples similaires.

#### **Solution :**

Commençons par construire la figure dans DR. GEO qui doit être comme ci-dessous :

FIG. 5.1 – Trapèze rectangle



La figure comprend les données à partir desquelles nous pouvons résoudre le problème. D'abord nous pouvons répondre à la première question de l'aire, pour cela nous pouvons écrire le script Scheme suivant ayant comme entrée les deux bases et la hauteur du trapèze :

```

define AB (getLength a1))
(define DC (getLength a2))
(define AD (getLength a3))
(/ ( * AD (+ AB DC )) 2 )

```

Nous calculons la longueur du segment  $BH$  en écrivant un script Scheme avec comme objet en entrée les segments  $AB$  et  $CD$ , et le texte du script est le suivant :

```

(define AB (getLength a1))
(define CD (getLength a2))
(- AB CD)

```

À ce stade, nous pouvons appliquer le théorème de Pythagore dans le triangle rectangle  $CHB$ . Ici aussi nous utilisons un script Scheme avec comme objets en entrée le segment  $CH$  et le script  $BH$  :

```

(define CH (getLength a1))
(define BH (getValue a2))
(+ (* CH CH) (* BH BH))

```

Finalement nous pouvons obtenir la valeur du segment  $BC$  en calculant la racine carrée de la valeur retournée par le script précédent :

```

(define q (getValue a1))
(sqrt q)

```

Les deux scripts précédents peuvent être regroupés en un seul script un peu plus élaboré.

Maintenant nous pouvons conclure l'exercice en calculant le périmètre avec un script Scheme :

```

(define AB (getLength a1))
(define CB (getValue a2))
(define DC (getLength a3))
(define AD (getLength a4))
(+ (+ AB CB ) (+ DC AD ))

```

## 5.2 Théorème et conjectures

Les scripts Scheme Section 3.2, page 28 permettent de résoudre des exercices mais aussi de comprendre de façon plus approfondie l'énonciation des théorèmes et de vérifier des conjectures.

Dans cette section nous commençons par analyser le théorème de Tolomeo :

*Étant donné un quadrilatère inscrit dans un cercle la somme du produit des côtés opposés est égale au produit des diagonales.*

Nous pouvons construire la figure avec DR. GEO comme ci-dessous

où nous avons implémenté deux scripts qui calculent respectivement la somme du produit des côtés opposés et le produit des diagonales.

Le premier script est le suivant :

```

(define AB (getLength a1))
(define DC (getLength a2))
(define BC (getLength a3))
(define AD (getLength a4))
(+ (* AB DC ) (* BC AD ))

```

Le second script est :

```

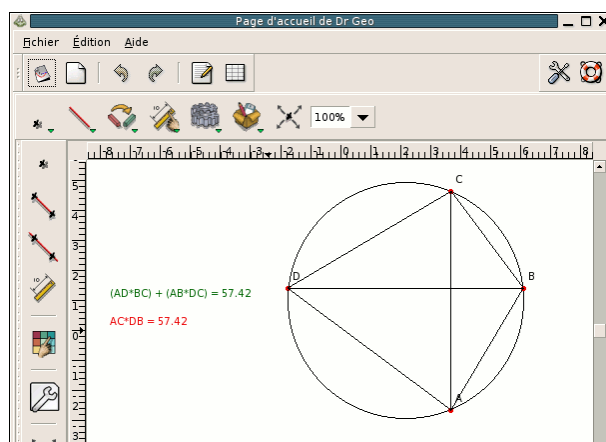
(define DB (getLength a1))
(define AC (getLength a2))
(* DB AC )

```



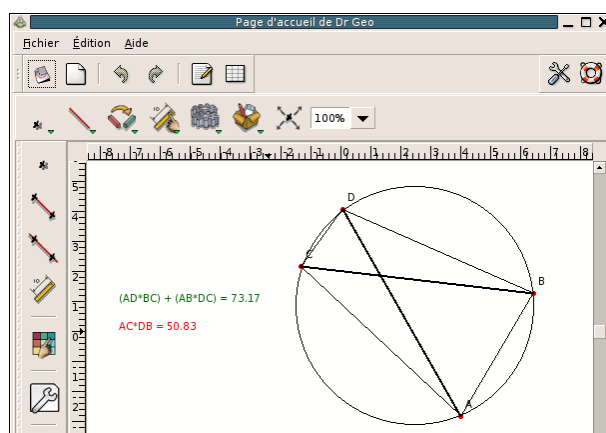


FIG. 5.2 – Théorème de Tolomeo : quadrilatère convexe



Comme nous pouvons le voir les valeurs retournées par les deux scripts, en accord avec le théorème de Tolomeo, sont les mêmes<sup>1</sup>. Lorsque nous modifions dynamiquement la figure, les valeurs des scripts sont toujours identiques, sauf dans la situation suivante :

FIG. 5.3 – Théorème de Tolomeo : quadrilatère non convexe



où le quadrilatère perd sa convexité.

Dans ce cas le théorème n'est pas vrai et l'énoncé précédent n'est pas bien énoncé, il doit donc être reformulé comme ci-dessous :

*Étant donné un quadrilatère CONVEXE inscrit dans un cercle la somme du produit des côtés opposés est égale au produit des diagonales.*

À ce stade les conjectures apparaissent naturellement : est-ce que le théorème de Tolomeo est valide pour un quadrilatère convexe non inscrit dans un cercle ?

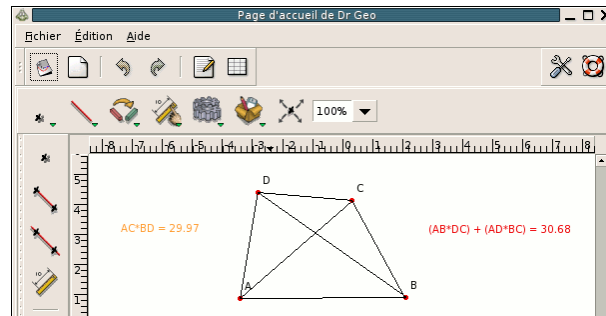
Avec DR. GEO nous pouvons vérifier que cette conjecture est fausse comme le montre la figure suivante :

Le lecteur n'aura pas de difficulté à utiliser DR. GEO dans la construction d'exemples didactiques, probablement plus connus, relativement aux théorèmes de Pythagore et d'Euclide.

<sup>1</sup> Il ne s'agit que d'une vérification numérique.



FIG. 5.4 – Réfutation de la conjecture

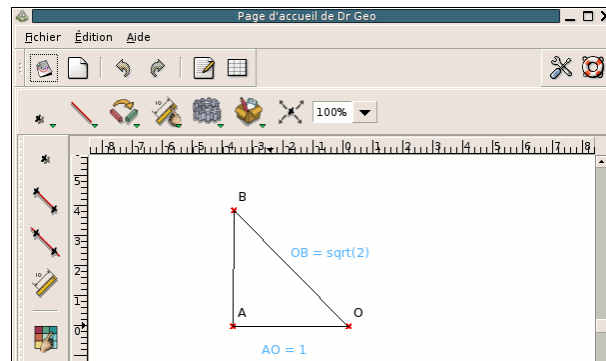


### 5.3 Nombre irrationnel

Une construction classique, relative au nombre irrationnel, connue sous le nom de spirale de Teodoro, permet de construire géométriquement la racine carrée de nombres entiers à partir d'un triangle rectangle isocèle.

Considérons le triangle  $OAB$  où  $OA = 1$  :

FIG. 5.5 – Construction de la racine de 2



Par le théorème de Pythagore nous avons  $OB$  égale à la racine carrée de 2. Si maintenant, avec la figure, nous construisons un nouveau triangle rectangle en  $B$ , avec les côtés  $OB$  et  $BC$  tel que  $BC = 1$ .

Toujours par le théorème de Pythagore, il est clair que l'hypothénuse  $OC$  de  $OBC$  a pour longueur la racine carrée de 3. En itérant le processus précédent à l'infini nous obtenons toutes les racines carrées des nombres naturels.

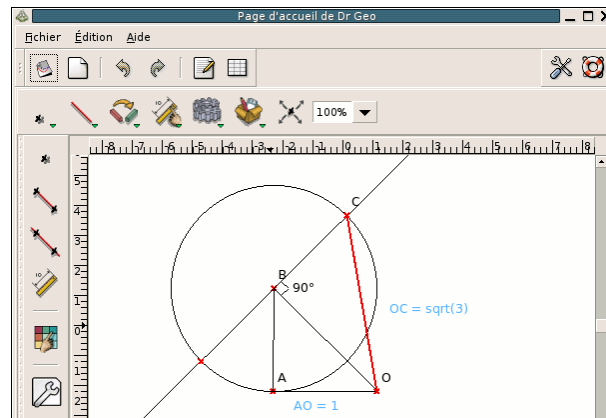
La nature itérative de la construction s'adapte parfaitement à l'utilisation des FSD. Considérons alors le code suivant :

```
(new-figure "Triangle")

(define (triangle p1 p2 p3 n)
  (let* ((s1 (Segment "" extremities p1 p2))
        (s2 (Segment "" extremities p2 p3))
        (s3 (Segment "" extremities p3 p1))
        (pe (Line "" orthogonal p3 s3))
        (ci (Circle "" center-segment p3 s2))
        (p4 (Point "" intersection2 pe ci)))
```



FIG. 5.6 – Construction de la racine 3

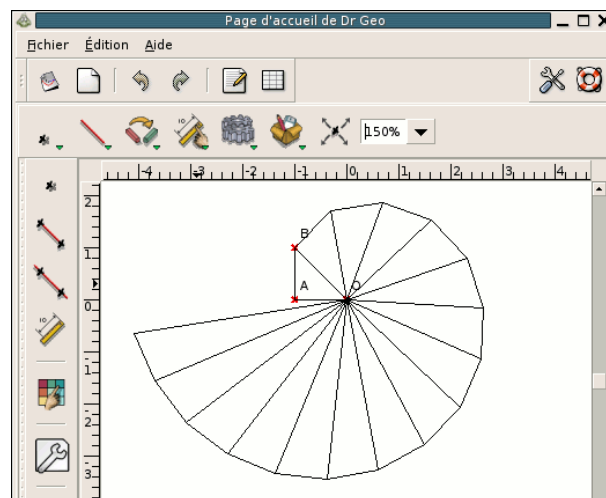


```
(send pe masked)
(send ci masked)
(send p4 masked)
(if (> n 0)
  (triangle p1 p3 p4 (- n 1))))
```

```
(lets Point "O" free 0 0)
(lets Point "A" free -1 0)
(lets Point "B" free -1 1)
(triangle O A B 15)
```

Le triangle du début est défini à travers les coordonnées seulement par commodité. Le code est la transcription littérale de la procédure itérative décrite précédemment. Une fois évalué par DR. GEO le code donne la figure suivante :

FIG. 5.7 – Spirale de Teodoro



Les hypothénuses de chaque triangle ont pour longueur les racines carrées des nombres entiers naturels compris entre 2 et 17.



## 5.4 Spirale de Baravelle

Comme nous l'avons vu précédemment, à l'aide de FSD il est possible de construire de façon intuitive et simple des figures permettant de *visionner* des situations qui en programmation sont récursives – ou cycliques.

Nous pouvons un peu approfondir cet aspect, en modifiant le code Scheme utilisé pour la construction des nombres irrationnels, afin d'obtenir une figure fameuse de la littérature des mathématiques, à savoir la spirale de Baravelle.

Le code Scheme définissant la spirale est le suivant :

```
(new-figure "Baravelle")

(define (triangle p1 p2 p3 n)
  (let* ((s1 (Segment "" extremities p1 p2))
        (s2 (Segment "" extremities p2 p3))
        (s3 (Segment "" extremities p3 p1))
        (m (Point "" middle-2pts p1 p3))
        (r (Segment "" extremities m p3))
        (pe (Line "" orthogonal p3 s3))
        (ci (Circle "" center-segment p3 r))
        (p4 (Point "" intersection2 pe ci)))
    (send pe masked)
    (send ci masked)
    (send p4 masked)
    (send m masked)
    (if (> n 0)
        (triangle m p3 p4 (- n 1)))))

(lets Point "A" free 0 5)
(lets Point "B" free 5 5)
(lets Point "C" free 5 0)
(triangle A B C 9)

(lets Point "D" free 0 -5)
(lets Point "E" free -5 -5)
(lets Point "F" free -5 0)
(triangle D E F 9)
```

À partir de la figure et du code Scheme correspondant nous percevons bien la nature itérative du mécanisme de construction de la figure. Un problème intéressant que nous laissons au lecteur, consiste à établir à quel moment les deux rameaux de la spirale convergent.

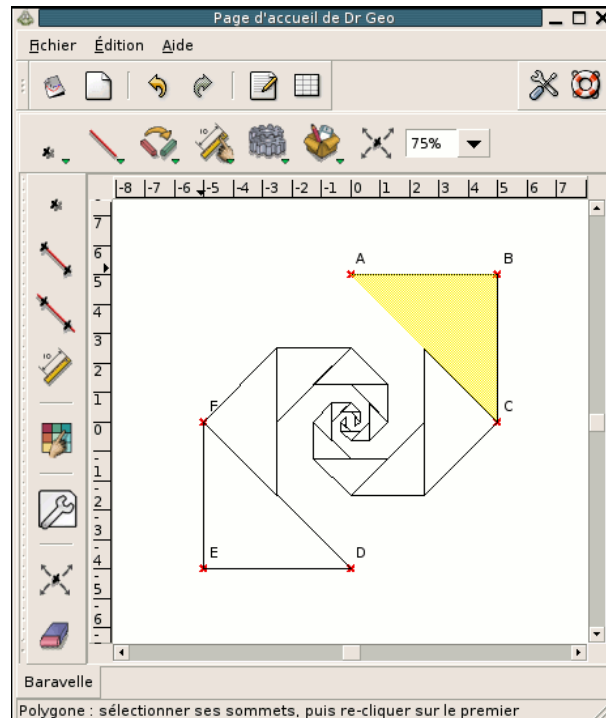
Une petite variation supplémentaire du code précédent :

```
(new-figure "Spirale")

(define (square p1 p2 p3 p4 n)
  (let* ((s1 (Segment "" extremities p1 p2))
        (s2 (Segment "" extremities p2 p3))
        (s3 (Segment "" extremities p3 p4))
        (s4 (Segment "" extremities p4 p1))
        (A (Point "" on-curve s1 1/10))
        (B (Point "" on-curve s2 1/10))
        (C (Point "" on-curve s3 1/10))
        (D (Point "" on-curve s4 1/10)))
    (send A masked)
```



FIG. 5.8 – La spirale de Baravelle suite à l'évaluation du code Scheme



```

(send B masked)
(send C masked)
(send D masked)
(if (> n 0)
  (square A B C D (- n 1))))

(lets Point "M" free 5 5)
(lets Point "N" free -5 5)
(lets Point "O" free -5 -5)
(lets Point "P" free 5 -5)

(square M N O P 30)

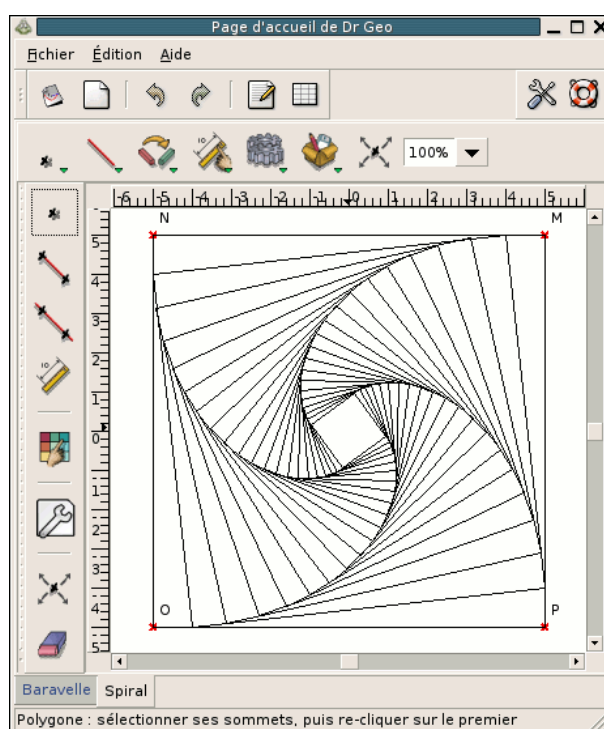
```

conduit à une spirale simplifiée.

Le lecteur est invité à se divertir en créant de nouvelles variations !



FIG. 5.9 – Spirale simplifiée

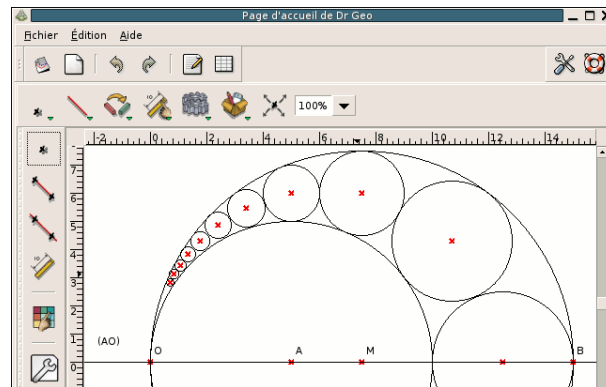


## 5.5 Catena di Pappo

Une utilisation de base de Figure Scheme de DR. GEO consiste en la reproduction de figure dont nous connaissons les caractéristiques analytiques.

L'exemple de construction que nous proposons est représenté par la fameuse "Catena di Pappo".

FIG. 5.10 – Catena di Pappo



Les centres et rayons successifs des cercles qui la constituent ont une expression analytique connue, il est donc aisé de reproduire la figure en tant de FSD.

```
(new-figure "Pappo")

(define (circle n)
  (let* (
    (r (Numeric "" free 0 0 (/ 15 (+ 6 (* n n)))))
    (c (Point "" free (* 5 (/ 15 (+ 6 (* n n)))
      (* 2 (* n (/ 15 (+ 6 (* n n)))))))
    (p (Circle "" center-radius c r)))
    (send r masked)
    (if (> n 0)
      (circle (- n 1)))))

(circle 8)

(lets Point "A" free 5 0)
(lets Point "O" free 0 0)
(lets Point "B" free 15 0)
(lets Point "M" middle-2pts B O)
(lets Circle "" 2points M O)
(lets Circle "" 2points A O)
(lets Line "" 2points A O)
```

le code de la figure est relativement intuitif et ne nécessite pas de commentaire.

Un exercice non trivial, que nous laissons au lecteur, consiste à déterminer une construction à la règle et au compas conduisant à une implémentation itérative.

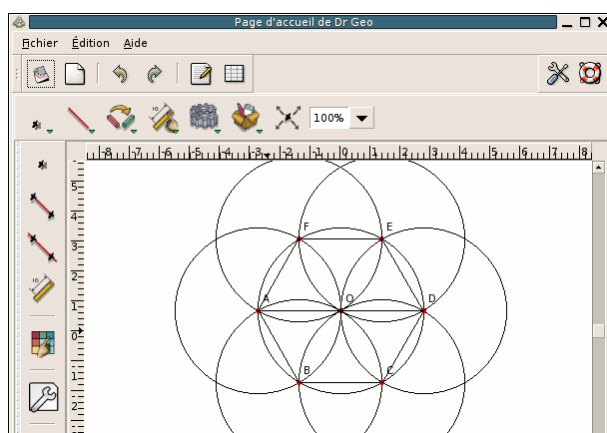


## 5.6 Calcul de $\pi$

Le calcul approximatif de  $\pi$  a joué un rôle important dans l'histoire des Mathématiques. Les méthodes pour ce type de calcul sont diverses et contiennent souvent des améliorations d'une méthode à l'autre. Nous vous proposons une approche du problème très simplifiée que nous appelons – même si ce n'est pas tout à fait exacte – **Méthode par exhaustion**. Cette approche a toutefois l'avantage de montrer l'essence même de la méthodologie.

Nous commençons avec la construction suivante d'un hexagone inscrit dans un cercle à partir du côté  $BC$ . Nous notons au passage qu'il est possible à partir de cette construction de construire et mémoriser une macro que nous pouvons nommer *Hexagone*.

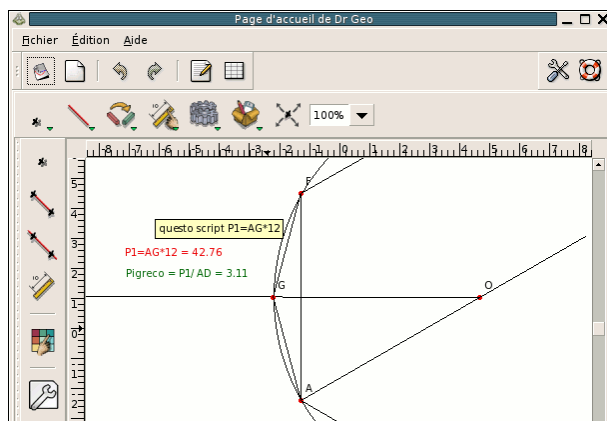
FIG. 5.11 – Hexagone régulier inscrit



L'idée de la méthode par exhaustion consiste dans un premier temps à approximer la longueur du cercle avec le périmètre  $P_0$  de l'hexagone et de calculer une approximation de  $\pi$  dividende de  $P_0$  par le diamètre du cercle. Clairement l'approximation de  $\pi$  obtenue sera de 3.

Lors d'une deuxième étape nous pouvons, en utilisant DR. GEO, construire un côté du dodécagone inscrit dans le cercle. Nous calculons son périmètre  $P_1$  ainsi qu'une approximation successive de  $\pi$  comme dividende de  $P_1$  par le diamètre.

FIG. 5.12 – Approximation de  $\pi$



En augmentant le nombre de côtés du polygone régulier inscrit nous obtenons de





meilleures approximations.





## Chapitre 6

# Astuces diverses

### 6.1 Créer un polygone régulier

DR. GEO peut créer des macro-constructions comportant des éléments de construction indépendants (i.e. des points libres, des valeurs libres, des scripts sans parent). Pour ce faire DR. GEO prend l'initiative de donner une ou des valeurs aux paramètres libres de ces objets<sup>1</sup>. La ou les valeur(s) choisie(s) sont en fait celle(s) de l'objet servant de modèle lors de la création de la macro.

Dans la suite, un petit exemple permettant de créer des polygones réguliers à  $n$  côtés est présenté. En plus des macro-constructions, un script Scheme intervient dans le processus.

L'exemple est guidé pas à pas mais le lecteur non familier avec les macro-constructions, les scripts ou plus généralement le fonctionnement de DR. GEO est fortement invité à lire les sections précédentes.

#### 6.1.1 Le modèle de polygone

Pour commencer, créer un point libre, puis un script sans parent – après avoir choisi l'outil script cliquer directement sur le fond de la figure. Éditer le script et lui donner comme valeur :

```
(/ (* 2 (acos -1)) 7)
```

Le résultat de ce script servira pour construire par rotation un polygone régulier à 7 côtés. Placer ensuite une valeur libre de valeur 2, ce sera le rayon de notre polygone.

Construire un cercle de centre le point et de rayon la valeur. Sur ce cercle placer un point. Nommer ce point  $O$ .

Maintenant utiliser l'outil rotation pour créer l'image de  $O$  par la rotation d'angle la valeur du script et de centre le centre du cercle. Continuer ensuite successivement jusqu'à obtenir les 7 sommets du polygone. Relier les sommets par des segments.

#### 6.1.2 La macro-construction du polygone

Ouvrir la boîte de dialogue de création de macro-construction et construire comme suit la macro :

1. **paramètre d'entrée** : le centre du cercle ;
2. **paramètres de sortie** : le point  $O$  et les 7 côtés du polygone.

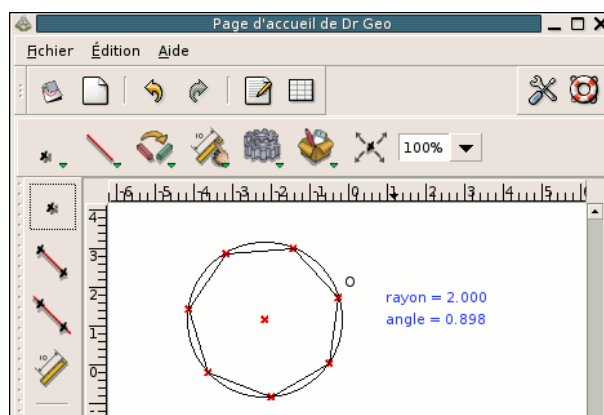
Voilà, la macro est prête.

Pour créer un polygone régulier à 7 côtés, ouvrir la boîte de dialogue d'exécution d'une macro, il suffit de sélectionner un point de la figure.

---

<sup>1</sup> Sinon, la macro-construction ne pourrait pas être instanciée. Par exemple quelle valeur donner à un objet de type valeur libre intervenant dans la séquence de construction ?

FIG. 6.1 – Résultat de la construction du polygone régulier à 7 côtés



### 6.1.3 Quelques considérations

Lorsque DR. GEO rencontre des éléments libres dans une macro-construction, il fixe leur valeur en prenant celle du modèle. Dans notre exemple, le script, la valeur et le point O sont de tels éléments libres. Lorsque la macro-construction est exécutée, ils sont fixés comme suit :

- la valeur à 2 ;
- le script à  $(/ (* 2 (\text{acos } -1)) 7)$  ;
- le point sur le cercle avec la même abscisse curviligne que O.

C'est ce dernier point qui donne tout son sens à l'utilisation conjointe des macro-constructions avec des scripts ou d'autres objets libres.

## 6.2 Imprimer une figure

Le plus simple est d'exporter la figure au format PostScript – Exportation PostScript Section 4.5.2, page 53. Utiliser ensuite un logiciel permettant de visionner des documents PostScript et imprimer le document. Le document imprimé de la sorte est de qualité vectorielle. Les logiciels GhostView, Gnome GhostView ou encore K GhostView conviennent parfaitement.

## 6.3 Placer un paragraphe de texte dans une figure

On peut avantageusement utiliser les scripts, d'une façon certes détournée. Se placer dans le mode de création de script – Outil Script Section 2.1.4, page 16 – cliquer directement sur le fond de l'écran. De cette façon un script sans paramètre d'entrée est créé.

Ensuite éditer les propriétés du script – Propriétés d'un objet Section 2.2.6, page 20. Dans la zone texte où le code Scheme est normalement écrit, placer un texte entre guillemets précédé de l'apostrophe, par exemple :

```
' "Animation et observations
=====
```

Déplace les points de la figure pour répondre aux questions suivantes.

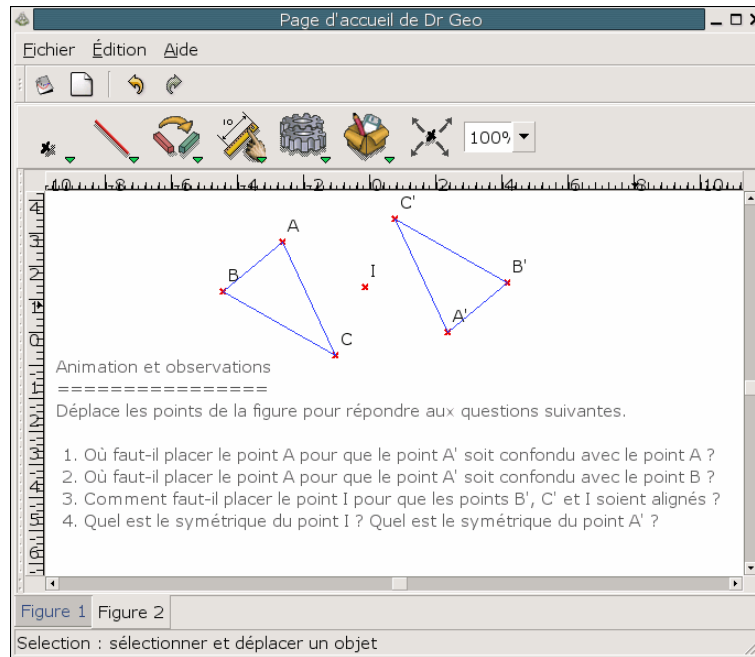
1. Où faut-il placer le point A pour que le point A' soit confondu avec le point A ?
2. Où faut-il placer le point A pour que le point A' soit confondu avec le point B ?
3. Comment faut-il placer le point I pour que les points B', C' et I soient alignés ?



### 6.3. PLACER UN PARAGRAPHE DE TEXTE DANS UNE FIGURE

4. Quel est le symétrique du point I ? Quel est le symétrique du point A' ?"

FIG. 6.2 – Exemple de paragraphe de texte dans une figure



Comme pour tout script, il est également possible de changer la couleur du texte.





# Table des figures

1.1	Écran de bienvenue de DR. GEO . . . . .	8
1.2	Une figure géométrique vide . . . . .	8
1.3	Une figure DR. GEO et sa description . . . . .	9
2.1	Catégories d'outils de DR. GEO et leur description . . . . .	11
2.2	Boîte de dialogue pour le style d'un objet point . . . . .	18
2.3	Boîte de dialogue pour le style d'un objet ligne . . . . .	18
2.4	Boîte de dialogue pour les nombres & les objets polygones . . . . .	19
2.5	Changement des coordonnées d'un point libre . . . . .	20
2.6	Modification d'une valeur libre . . . . .	20
2.7	Modification d'un script . . . . .	20
2.8	Préférences des figures géométriques . . . . .	21
2.9	Changement du nom d'une figure . . . . .	22
3.1	La figure initiale . . . . .	24
3.2	La figure avec la construction finale . . . . .	24
3.3	Première page de la boîte de dialogue de l'assistant pour construire une macro- construction . . . . .	25
3.4	La seconde page, les trois points sont sélectionnés . . . . .	25
3.5	La troisième page, le cercle et son centre sont sélectionnés . . . . .	25
3.6	La quatrième page, le nom et la description de la macro-construction . . . . .	26
3.7	L'utilisateur sélectionne les paramètres d'entrée dans la figure . . . . .	26
3.8	Une figure avec trois points . . . . .	27
3.9	La figure finale avec le cercle et son centre . . . . .	27
3.10	Exécuter une macro-construction directement depuis le menu <b>Macro-constructions</b> . . . . .	27
3.11	La figure que nous obtiendrons . . . . .	30
3.12	Un polygone régulier à 15 côtés . . . . .	46
3.13	Une courbe fractale simulant la représentation d'un arbre . . . . .	48
3.14	La boîte dialogue pour verrouiller des outils. . . . .	49
3.15	La boîte de dialogue pour déverrouiller l'interface d'une figure. . . . .	49
4.1	La boîte dialogue de session DR. GEO . . . . .	52
4.2	Une figure avec une zone d'exportation définie . . . . .	54
4.3	La zone d'exportation a été exportée dans un document PostScript, il est visualisé avec GhostView . . . . .	54
5.1	Trapèze rectangle . . . . .	55
5.2	Théorème de Tolomeo : quadrilatère convexe . . . . .	57
5.3	Théorème de Tolomeo : quadrilatère non convexe . . . . .	57
5.4	Réfutation de la conjecture . . . . .	58
5.5	Construction de la racine de 2 . . . . .	58
5.6	Construction de la racine 3 . . . . .	59

## TABLE DES FIGURES

---

5.7	Spirale de Teodoro . . . . .	59
5.8	La spirale de Baravelle suite à l'évaluation du code Scheme . . . . .	61
5.9	Spirale simplifiée . . . . .	62
5.10	Catena di Pappo . . . . .	63
5.11	Hexagone régulier inscrit . . . . .	64
5.12	Approximation de $\pi$ . . . . .	64
6.1	Résultat de la construction du polygone régulier à 7 côtés . . . . .	68
6.2	Exemple de paragraphe de texte dans une figure . . . . .	69





# Index

- Afficher
  - Texte, 68
- Aide
  - Préférences par défaut
  - Visionneuse, 21
- Angle
  - Géométrie, 15
  - Orienté, 15
- Arc de cercle, 13
  - Longueur, 15
- Cercle, 13
  - Périmètre, 15
- Demi-droite, 13
- Droite, 13
  - Distance, 15
  - Parallèle, 14
  - Pente, 15
  - Perpendiculaire, 14
- Editer
  - Point, 20
  - Préférences par défaut, 21
    - Annuler/Refaire, 21
    - Noms, 21
  - Propriétés, 20
  - Script, 20
  - Style, 17
  - Style par défaut, 21
  - Valeur, 20
- Figure
  - Afficher
    - Grille, 21
    - Texte, 68
  - Déplacer, 17
  - Enregistrer, 51
  - Exporter
    - L<sup>A</sup>T<sub>E</sub>X, 52
    - Définir la zone d'exportation, 53
    - Fly Draw/WIMS, 53
    - Image PNG, 53
    - PostScript, 53
  - Imprimer, 68
  - Ouvrir, 52
  - Renommer, 22
- Figure Scheme de DR. GEO
  - Création d'objets, 36
    - Angle, 43
    - Arc de cercle, 39
    - Attributs des objets, 43
    - Cercle, 39
    - Demi-droite, 38
    - Droite, 38
    - Lieu, 41
    - Nombre, 41
    - Point, 37
    - Polygone, 40
    - Segment, 38
    - Transformations géométriques, 40
    - Vecteur, 41
  - Evaluer, 34
  - Exemples, 34, 45
    - Catena di Pappo, 63
    - Fractal, 46
    - Polygone régulier, 46
    - Spirale de Baravelle, 60
    - Spirale de Teodoro, 58
  - Introduction, 34
  - Nouvelle figure, 34
  - Synonymes, 44
- Grille
  - Afficher, 21
- Historique
  - Arbre logique, 16
- Imprimer
  - Figure, 68
- Lieu géométrique, 13
  - Script, 30
- Macro-construction, 16
  - Créer, 24
  - Enregistrer, 51, 52
  - Exécuter, 26
    - Boîte de dialogue, 26
  - Menu, 26
  - Introduction, 23

- Ouvrir, 52
- Polygone, 67
- Script, 67
- Nombre, 15
  - Valeur libre, 15
- Objet
  - Déplacer, 17
  - Masquer, 17
  - Renommer, 17
  - Supprimer, 17
- Outils
  - Déverrouiller, 48
  - Verrouiller, 47
- Point
  - Coordonnées, 16
  - Défini par coordonnées, 12
  - Intersection, 12
  - Libre, 12
  - Milieu, 12
  - Renommer, 17
- Polygone, 14
  - Macro-construction, 67
- Renommer
  - Objet, 17
- Script, 16
  - getAbscissa, 31
  - getAngle, 33
  - getCenter, 33
  - getCoordinates, 31
  - getLength, 32, 33
  - getNorm, 32
  - getRadius, 33
  - getSlope, 32
  - getUnit, 32
  - getValue, 33
  - move, 34
  - random, 29
  - setAbscissa, 31
  - setCoordinates, 32
  - setValue, 33
- Exemples, 28
  - Aire d'un trapèze, 55
  - Différence de longueur de segments, 56
  - Périmètre, 56
  - Racine carrée, 56
  - Tangente à une courbe, 30
  - Théorème de Pythagore, 56
  - Théorème de Tolomeo, 56
- Intervalle, 30
- Introduction, 28
- Paramètres, 29
- Segment, 13
  - Longueur, 15
- Session
  - Enregistrer, 51
  - Ouvrir, 52
- Transformation
  - Figure Scheme de DR. GEO, 40
  - Homothétie, 15
  - Rotation, 15
  - Symétrie
    - Axiale, 14
    - Centrale, 14
  - Translation, 14
- Vecteur, 13
  - Coordonnées, 16
  - Norme, 15



## **Annexe A**

# **Historique**

### **Rédactions originales de la documentation**

Hilaire Fernandes (anglais et français), Andrea Centomo (italien).

### **Traductions vers le français**

Jean-Philippe Georget, Hilaire Fernandes.

### **Relectures**

François Audirac, Odile Benassy, Gérard Blanchet, Guy Veyssière.



## Annexe B

# GNU Free Documentation License

GNU Free Documentation License  
Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

### 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple



---

HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on



the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

#### 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications





- adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
  - G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
  - H. Include an unaltered copy of this License.
  - I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
  - J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
  - K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
  - L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
  - M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
  - N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
  - O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of,



you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit.



---

When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number.



If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c)  YEAR  YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

