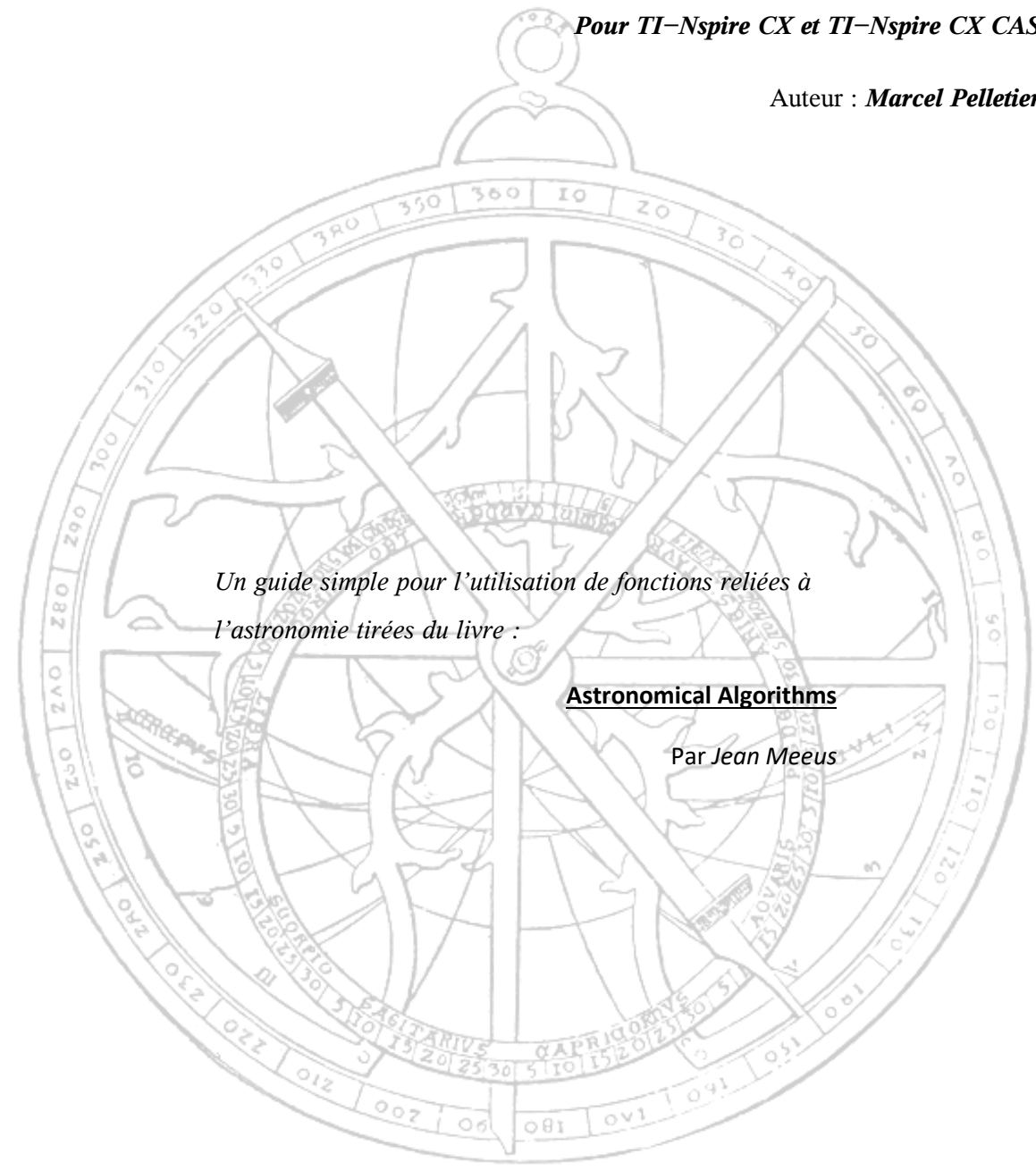


AstroLab_5

Pour TI-Nspire CX et TI-Nspire CX CAS

Auteur : **Marcel Pelletier**



*Un guide simple pour l'utilisation de fonctions reliées à
l'astronomie tirées du livre :*

Astronomical Algorithms

Par Jean Meeus

Voir un univers dans un grain de sable.

Un paradis dans une fleur sauvage.

Tenir l'infini dans la paume de sa main.

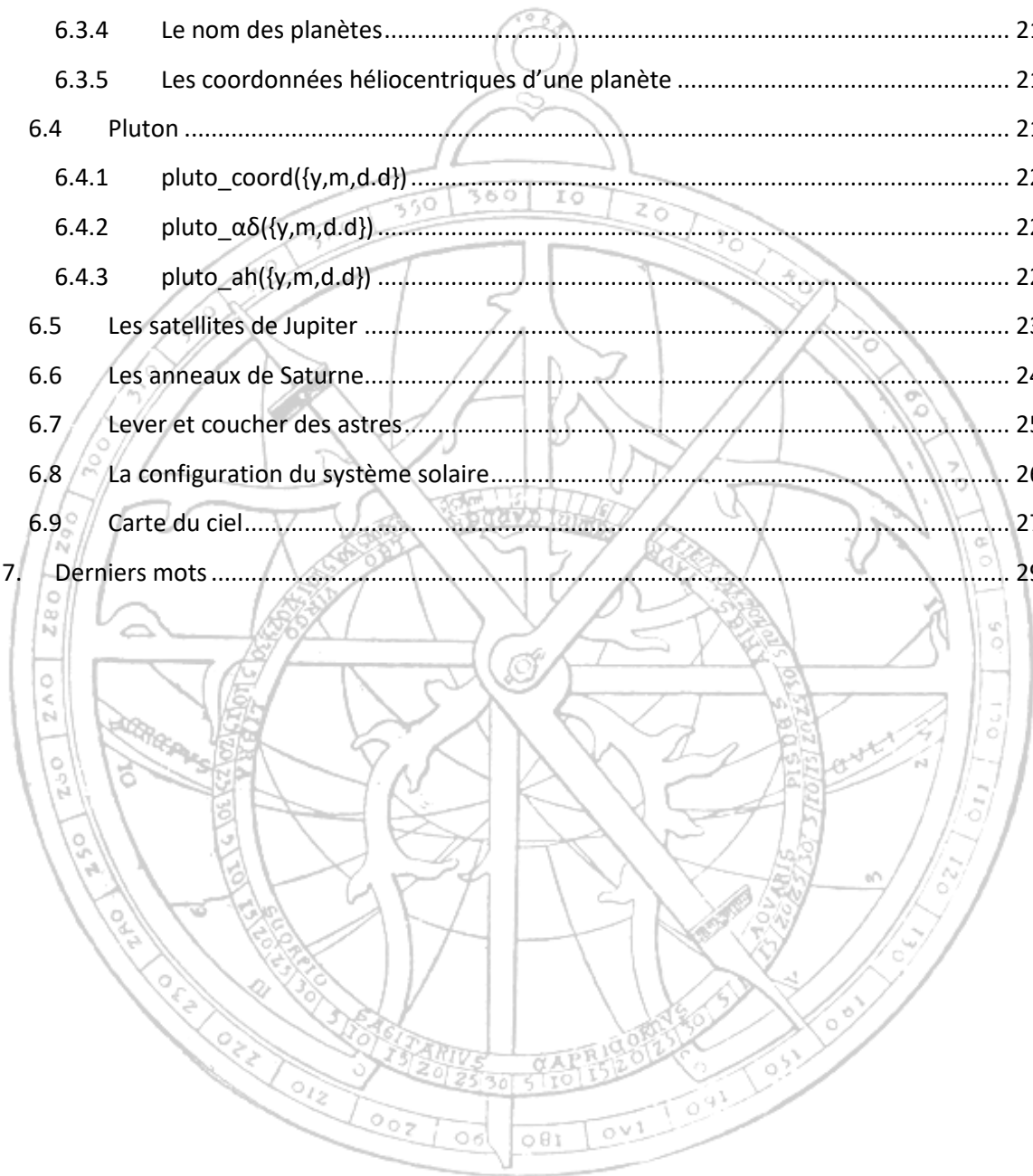
Et l'éternité dans une heure...

William Blake

Table des matières

1.	Historique de création.....	4
2.	Installation.....	5
3.	Prise en main initiale	6
4.	Initialisation	7
5.	Les fonctions de calendrier	8
5.1	makedatelshort({y,m,d,h,m,s}).....	8
5.2	makedatelist({y,m,d,d})	8
5.3	julianday({y,m,d,d})	8
5.4	datefromjd(JD).....	9
5.5	dayoftheweek({y,m,d,d}).....	9
5.6	print_date({y,m,d,d}).....	9
5.7	delta_days({y,m,d,d},{y,m,d,d}).....	10
5.8	datexdaysafter({y,m,d,d},X).....	10
5.9	easter_date(y)	10
5.10	vendredi_13(y)	10
5.11	touniversaltime({y,m,d,d})	11
5.12	listnow({y,m,d,h,m,s})	11
6.	Les fonctions reliées à l'astronomie.....	12
6.1	Le Soleil.....	12
6.1.1	solarcoordinate({y,m,d,d})	12
6.1.2	sun_αδ({y,m,d,d}).....	13
6.1.3	sun_ah({y,m,d,d})	14
6.1.4	seasons(y, heure avancée)	14
6.2	La Lune.....	15
6.2.1	moon_coord({y,m,d,d})	15
6.2.2	moon_αδ({y,m,d,d})	16
6.2.3	moon_ah({y,m,d,d})	17
6.2.4	illu_frac_moon({y,m,d,d})	17
6.2.5	moon_illu_view({y,m,d,d})	18
6.2.6	phases_moon({y,m,d,d},heure avancée)	18

6.3	Les planètes.....	19
6.3.1	planet_coord(object,{y,m,d.d})	19
6.3.2	planet_αδ(object,{y,m,d.d})	20
6.3.3	planet_ah(object,{y,m,d.d})	20
6.3.4	Le nom des planètes.....	21
6.3.5	Les coordonnées héliocentriques d'une planète	21
6.4	Pluton	21
6.4.1	pluto_coord({y,m,d.d})	22
6.4.2	pluto_αδ({y,m,d.d})	22
6.4.3	pluto_ah({y,m,d.d})	22
6.5	Les satellites de Jupiter	23
6.6	Les anneaux de Saturne.....	24
6.7	Lever et coucher des astres.....	25
6.8	La configuration du système solaire.....	26
6.9	Carte du ciel.....	27
7.	Derniers mots	29



1. Historique de création

Ma première calculatrice à donner la position des planètes était une **TI-58C** de *Texas Instruments* avec son module de navigation. L'almanach nautique était par contre nécessaire pour l'utilisation de certains programmes de ce module d'extension mémoire. Par la suite, à l'aide du **HP-41CX** de *Hewlett-Packard* aussi équipé du module de navigation, j'obtenais le même résultat sans l'aide de l'almanach nautique!

À partir de ce moment, j'ai toujours voulu que mes calculatrices ou ordinateurs soient en mesure de me donner la position des planètes sans l'aide d'un almanach.

Lorsque j'étais étudiant en physique à l'Université de Montréal (1983-1986,) j'ai parcouru cet ouvrage de Jean Meeus : [Astronomical formulae for calculators](#) (1979). Les calculatrices de ce moment étaient lentes et ne possédaient que quelques kilooctets de mémoire vive. La technique décrite dans ce livre convenait parfaitement aux calculatrices de ce moment.

En 1991, avec la publication de [Astronomical Algorithms](#), la méthode proposée change. Il est maintenant possible d'obtenir les principaux paramètres des orbites planétaires à partir de séries contenant un certain nombre de termes. Ceci demande une calculatrice plus performante et ayant plus de mémoire.

Dans les années 1990-2000, toutes mes calculatrices ou *palmtop* avaient ces caractéristiques.

Parmi cette liste, **HP-48S/SX**, **HP-48G**, **HP-28C/S**, **PC-1500**, **HP-100LX**, **HP-200LX**, **TI-89 Titanium**, **HP-49G**, j'ai choisi la **TI-89 Titanium** pour **Astro Lab 1**. Cette première version se limitait à donner l'altitude et l'azimut des principales planètes.

Astro Lab 2 sur **TI-Nspire CX**, était une version plus complète mais avec une programmation trop complexe. Beaucoup de petites fonctions compliquaient inutilement l'application. J'ai corrigé ce défaut pour mieux faire avec **Astro Lab 3** sur **Mathematica**, encore disponible sur le forum de **Mathematica** : http://community.wolfram.com/groups/-/m/t/315991?p_p_auth=vVgO5KYw

La version la plus complète à ce jour est **Astro Lab 4** sur la **HP Prime**. Encore disponible sur le site **HP Museum**: <http://www.hpmuseum.org/forum/thread-4246.html>. La vitesse de cette calculatrice et ses possibilités graphiques sont parfaites pour cette application.

Enfin, **AstroLab_5** sur **TI-Nspire CX** ou **TI-Nspire CX CAS**.

La maturité du système d'exploitation de cette calculatrice me laisse croire à une évolution matérielle très prochaine de la calculatrice. Elle sera beaucoup plus rapide, aura une horloge interne ainsi que des capacités graphiques équivalentes ou supérieures à la **HP Prime**. Pour pouvoir inclure toutes ces nouvelles fonctionnalités à **AstroLab_5**, j'ai décidé de développer l'application en deux temps. **AstroLab_5** comporte toutes les fonctions de calculs et **AstroLab_5.5** sera ma version la plus complète, incluant cette fois des programmes d'entrées de données et de sorties des résultats en mode graphique.

2. Installation

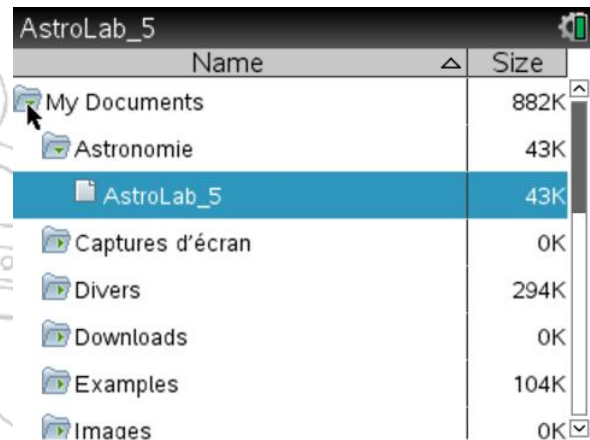
Pour installer **AstroLab_5** sur la **TI_Nspire**, il suffit d'y déposer ou coller le dossier *Astronomie*. Il n'y a pas de librairie donc nul besoin de *rafraichir* les librairies.

Après l'ouverture du dossier *Astronomie*, l'application est visible.

Il suffit alors de choisir le document **AstroLab_5**.

Une fois ouvert, le document a *sept onglets*.

Le premier onglet contient une courte description de toutes les fonctions disponibles dans l'application, tandis que le deuxième correspond à la page de calculs principale d'**AstroLab_5**.



L'onglet de la page d'informations.

L'onglet de la page de calculs.



La première fonction dans la liste alphabétique des fonctions est :

apparent_s_time({y, m, d. d})

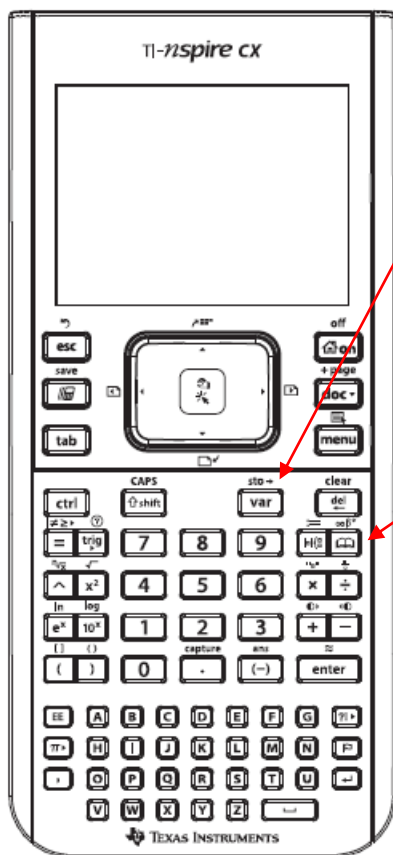
Le **nom** de la fonction provient de sa description anglaise, soit *apparent sidereal time*.

L'**argument** de la fonction $\{y, m, d. d\}$ est la *liste de date courte* ou *short date list* en anglais.

Cette façon d'écrire le moment est l'argument principal de la majorité des fonctions d'**AstroLab_5**.

3. Prise en main initiale

Pour bien utiliser **AstroLab_5**, deux techniques permettent de choisir une fonction donnée.

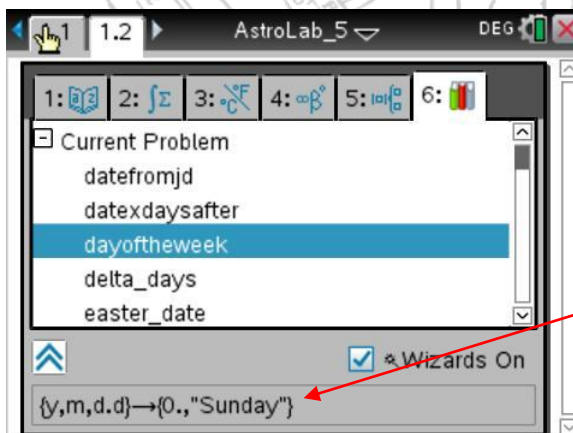
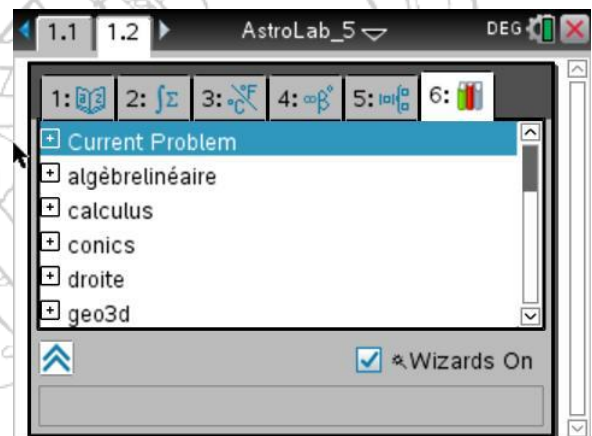


Première technique : La touche **var**

Permet d'obtenir tout le contenu du dossier (variables, fonctions et programmes) par ordre alphabétique.

Seconde technique : La touche **catalogue**

Permet d'obtenir de l'aide sur l'ensemble des fonctions de la **TI-Nspire**.
Le sixième onglet donne la liste des librairies. Pour **AstroLab_5**, il faut développer le *problème courant*.



Les fonctions ou programmes publics d'**AstroLab_5** apparaissent par ordre alphabétique et en plus, au bas de l'écran se trouve la forme de l'argument ainsi que la forme du résultat de la fonction.

4. Initialisation

Pour initialiser **AstroLab_5**, il faut définir le lieu d'observation, le fuseau horaire, l'heure d'été et la différence entre le temps dynamique et le temps universel, soit $\Delta T = DT - UT$.

initialisation()

Cette fonction permet de sauvegarder ces données dans la variable **lieu**.

Remarque

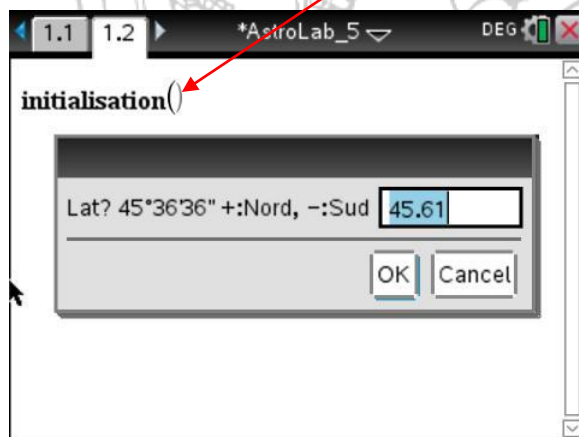
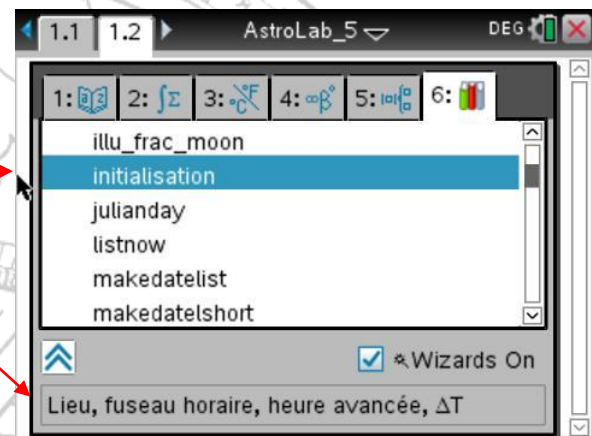
Pour plusieurs des exemples de ce guide, le contenu de cette variable doit-être celui proposé.

Il faut maintenant lancer la fonction d'initialisation!

Choisir cette fonction!

Cette fonction est en fait un programme, qui demande à l'utilisateur d'entrer ces données.

Il n'y a pas cette fois-ci d'argument, donc il ne faut rien inscrire entre les parenthèses.



Il est possible d'entrer la longitude ou la latitude en (°, ', ") ou en degrés décimaux. La dernière sauvegarde est proposée sous les deux formes. Pour entrer les secondes, il faut entrer **deux fois le caractère ' .** Donc 25 sec = 25''.

Pour voir le contenu sauvegardé, il suffit de taper **lieu**. Ce lieu correspond aux coordonnées de ma demeure, à Montréal.

5. Les fonctions de calendrier

Il y a douze fonctions de calendrier dans **AstroLab_5**. Plusieurs de ces fonctions donnent un résultat qui peut devenir l'argument d'une autre fonction. Ceci permet d'éviter d'avoir à entrer une grande quantité de données avec un très petit clavier.

5.1 makedatelshort{(y,m,d,h,m,s)}

Cette fonction permet de créer une liste de date courte.

Exemple

La liste de date courte du 28 juin 2018 à 14h43m23s sera {2018,6,28.6134606481}.



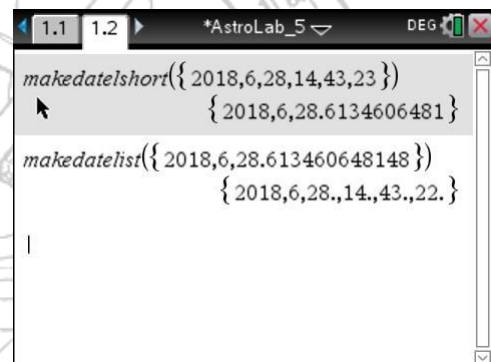
```
makedatelshort({ 2018,6,28,14,43,23 })
{ 2018,6,28.6134606481 }
```

5.2 makedatelist{(y,m,d.d)}

Cette fonction permet d'obtenir la liste de date longue correspondant à une liste de date courte.

Exemple

À partir du résultat précédent, cette fonction redonne la date correspondant au 28 juin 2018 à 14h43m23s.



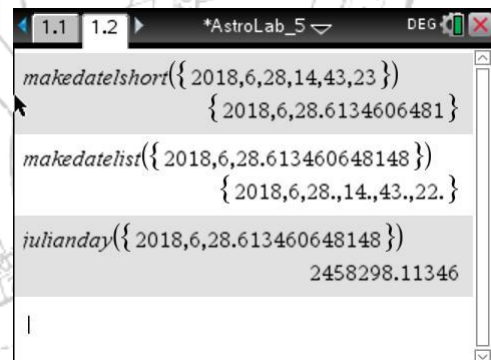
```
makedatelshort({ 2018,6,28,14,43,23 })
{ 2018,6,28.6134606481 }
makedatelist({ 2018,6,28.613460648148 })
{ 2018,6,28.,14.,43.,22. }
```

5.3 julianday{(y,m,d.d)}

Cette fonction renvoie le jour julien correspondant à la liste de date courte donnée en argument.

Exemple

Le jour julien correspondant au 28 juin 2018 à 14h43m23s sera 2458298.11346.



```
makedatelshort({ 2018,6,28,14,43,23 })
{ 2018,6,28.6134606481 }
makedatelist({ 2018,6,28.613460648148 })
{ 2018,6,28.,14.,43.,22. }
julianday({ 2018,6,28.613460648148 })
2458298.11346
```


5.4 datefromjd(JD)

Cette fonction renvoie la liste de date courte correspondant au jour julien donné.

Exemple

La liste de date courte correspondant au jour julien précédent est {2018,6,28.6134606}

```
1.1 1.2 *AstroLab_5 DEG
[ 2018,6,28.6134606 ]
makedatelist({ 2018,6,28.613460648148 })
{ 2018,6,28.,14.,43.,22. }
julianday({ 2018,6,28.613460648148 })
2458298.11346
datefromjd(2458298.1134606)
{ 2018.,6.,28.6134606 }
```

Les fonctions peuvent naturellement devenir l'argument d'une autre fonction. Pour l'exemple précédent, nous pourrions passer directement à la liste de date longue comme suit :

```
makedatelist(datefromjd(2458298.1134606))
```

Nous obtenons alors le 28 juin 2018, à 14h43m22s.

```
1.1 1.2 *AstroLab_5 DEG
[ 2018,6,28.6134606 ]
julianday({ 2018,6,28.613460648148 })
2458298.11346
datefromjd(2458298.1134606)
{ 2018.,6.,28.6134606 }
makedatelist(datefromjd(2458298.1134606))
{ 2018.,6.,28.,14.,43.,22. }
```

5.5 dayoftheweek({y,m,d.d})

À partir d'une liste de date courte, cette fonction donne le jour de la semaine. Pour cette fonction, dimanche correspond au jour zéro.

Exemple

Le jour de la semaine correspondant à la liste de date courte {2018,6,28.6134606} est *Jeudi*.

```
1.1 1.2 *AstroLab_5 DEG
[ 2018,6,28.6134606 ]
datefromjd(2458298.1134606)
{ 2018.,6.,28.6134606 }
makedatelist(datefromjd(2458298.1134606))
{ 2018.,6.,28.,14.,43.,22. }
dayoftheweek({ 2018.,6.,28.6134606 })
{ 4., "Thursday" }
```

5.6 print_date({y,m,d.d})

Plusieurs fonctions en astronomie donnent comme résultat un jour julien qui doit alors être transformé selon la langue. La fonction *print_date*({y,m,d.d}) donne une chaîne de caractères donnant la date, le jour et l'heure correspondant à la liste de date courte.

```
1.1 1.2 *AstroLab_5 DEG
[ 2018,6,28.6134606 ]
makedatelist(datefromjd(2458298.1134606))
{ 2018.,6.,28.,14.,43.,22. }
dayoftheweek({ 2018.,6.,28.6134606 })
{ 4., "Thursday" }
print_date({ 2018.,6.,28.6134606 })
"Thursday June 28, 2018. 14:43:23"
```

5.7 delta_days({y,m,d.d},{y,m,d.d})

Cette fonction donne le nombre de jours entre deux dates.

Exemple

Combien de jours y a-t-il entre le 1^{er} janvier 2000 et le 28 juin 2018?

Il y a 6753 jours entre ces deux dates.

```
1.1 1.2 *AstroLab_5 DEG
{ 2018,6,28,14,43,23 }
dayoftheweek({ 2018,6,28.6134606 })
{ 4, "Thursday" }
print_date({ 2018,6,28.6134606 })
"Thursday June 28, 2018. 14:43:23"
delta_days({ 2000,1,1 }, { 2018,6,28 })
6753.
```

5.8 datexdaysafter({y,m,d.d},X)

Cette fonction renvoie la date du $x^{\text{ième}}$ jour après la liste de date courte donnée comme argument.

Exemple

Quelle est la date du 6753^e jour après le 1^{er} janvier 2000?

Il s'agit du 28 juin 2018.

```
1.1 1.2 *AstroLab_5 DEG
{ 2, "Thursday" }
print_date({ 2018,6,28.6134606 })
"Thursday June 28, 2018. 14:43:23"
delta_days({ 2000,1,1 }, { 2018,6,28 })
6753.
datexdaysafter({ 2000,1,1 }, 6753)
{ 2018,6,28. }
```

5.9 easter_date(y)

Renvoie la date du Dimanche de Pâques.

Exemple

En 2018, Pâques sera le 1^{er} avril.

```
1.1 1.2 *AstroLab_5 DEG
print_date({ 2018,6,28.6134606 })
"Thursday June 28, 2018. 14:43:23"
delta_days({ 2000,1,1 }, { 2018,6,28 })
6753.
datexdaysafter({ 2000,1,1 }, 6753)
{ 2018,6,28. }
easter_date(2018)
{ 2018,4,1 }
```

5.10 vendredi_13(y)

Ce petit programme renvoie tous les vendredis 13 de l'année donnée. Il est facile d'inclure les fonctions de calendrier d'**AstroLab_5** dans d'autres programmes.

```
1.1 1.2 *AstroLab_5 DEG
{ 2018,6,28. }
easter_date(2018)
{ 2018,4,1 }
vendredi_13(2018)
Friday April 13, 2018 00:00:00
Friday July 13, 2018 00:00:00
Done
```

5.11 touniversaltime({y,m,d,d})

Cette fonction transforme la liste de date courte en une autre correspondant au temps universel de ce moment. Pour ce faire, la fonction utilise le fuseau horaire du lieu d'observation contenu dans la variable **lieu**, ainsi que l'indicateur d'heure avancée.


Cette fonction est nécessaire pour les fonctions de calculs astronomiques.

Exemple

Quelle est la liste de date courte correspondant au 7 septembre 2017 à 19h30m00s?

Pour mon lieu d'observation (Montréal), il faut ajouter 5 heures mais étant donné l'heure avancée d'été, il ne faut ajouter que 4 heures.

L'heure universelle à ce moment sera 23h30m00s du même jour.



```
makedatelshort({ 2017,9,7,19,30,0 })
{ 2017,9,7.8125 }

touniversaltime({ 2017,9,7.8125 })
{ 2017.,9.,7.9791667 }

makedatelist({ 2017.,9.,7.9791667 })
{ 2017.,9.,7.,23.,30.,0. }
```

5.12 listnow({y,m,d,h,m,s})

Pour l'instant, cette fonction est identique à la fonction **makedatelshort**. L'utilisateur doit donc fournir en argument, la liste de date longue du moment présent. La prochaine génération de la **TI-Nspire** aura probablement une horloge interne, ce qui permettra à la fonction de donner la liste de date courte correspondant à l'heure et à la date de l'horloge interne sans que l'utilisateur ait à fournir ces données.

6. Les fonctions liées à l'astronomie

Toutes les fonctions de calculs astronomiques ont comme argument une *liste de date courte* correspondant au *temps universel* ou au *temps terrestre ou dynamique* (avec $\Delta T = 0$).

La fonction ***touniversaltime*** est donc employée de façon courante pour la conversion du temps local au temps universel.

Attention

La liste ci-contre de la variable ***lieu***, est utilisée pour plusieurs des exemples qui suivent.



6.1 Le Soleil

Les fonctions liées au Soleil sont

- *sun_ah*({y,m,d.d}) : Renvoie l'azimuth et l'altitude du Soleil;
- *sun_αδ*({y,m,d.d}) : Renvoie l'ascension droite et la déclinaison du Soleil;
- *seasons*(y, heure avancée) : Renvoie la date et l'heure du début des saisons;
- *solarcoordinate*({y,m,d.d}) : Renvoie la longitude, la longitude apparente, la latitude apparente, l'ascension droite, la déclinaison et la distance Soleil-Terre. Cette fonction est disponible avec la touche **var**.

6.1.1 *solarcoordinate*({y,m,d.d})

Cette fonction renvoie une liste contenant les coordonnées du Soleil pour la liste de date courte fournie en argument et **correspondant au temps universel (TU)**. La liste obtenue n'est pas transformée, elle contient des valeurs numériques qui peuvent ensuite être réutilisées par une autre fonction ou un autre programme.

Exemple

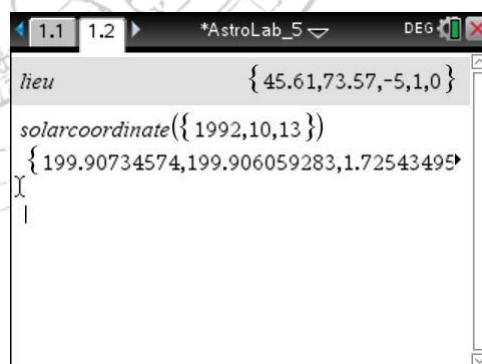
Calculez la position du Soleil le 13 octobre 1992, à 0h **TD**.

Puisque $TD = TU + \Delta T$, il faut réinitialiser **AstroLab_5** avec $\Delta T = 0$.

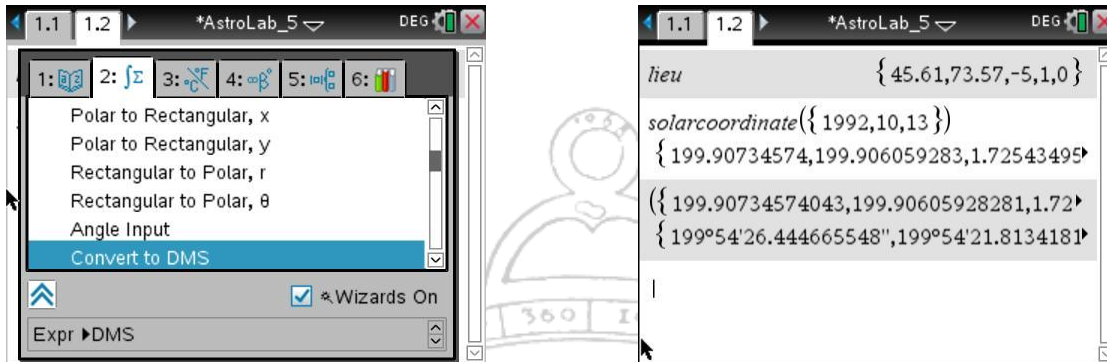
① *initialisation*()

② *solarcoordinate*({1992,10,13})

La liste obtenue contient : { $\theta^\circ, \lambda^\circ, \beta^\circ, \alpha_{\text{heure}}, \delta^\circ, R_{\text{au}}$ }

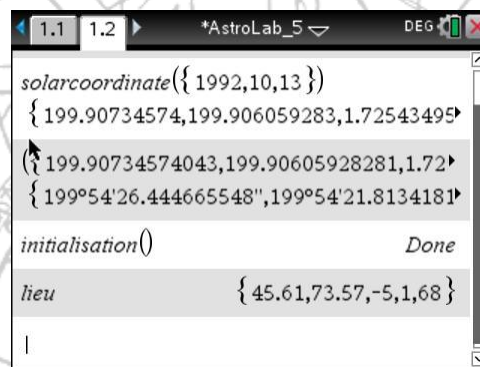


③ Pour convertir rapidement la liste sous la forme *DMS*, il est possible de se servir de cette commande accessible à partir de l'onglet 2 du **catalogue**.



Il faut naturellement se souvenir que *R* n'est pas un angle, mais une distance!

④ Réinitialiser **Astrolab_5** avec $\Delta T = 68$ secondes!



6.1.2 $\text{sun_}\alpha\delta(\{y,m,d,d\})$

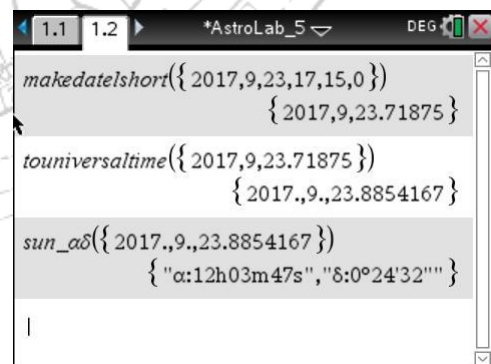
Pour obtenir plus rapidement l'ascension droite et la déclinaison du Soleil avec le format adéquat, il est préférable d'utiliser cette fonction.

Exemple

Quelle est la position du Soleil le 23 septembre 2017, à 17h15m HAE ?

Dans cet exemple, la liste *lieu* est appropriée.

- ① Création de la liste de date courte;
- ② Transformer la liste au temps universel;
- ③ Calculer la position du Soleil.



6.1.3 sun_ah({y,m,d.d})

Pour obtenir l'azimut et l'altitude du Soleil, au moment donné dans l'exemple précédent, il suffit d'utiliser l'historique pour reprendre la liste de date courte correspondant au temps universel de ce moment.

```

1.2 *AstroLab_5 DEG
makedatelshort({ 2017,9,23,17,15,0 })
    { 2017,9,23.71875 }
touniversaltime({ 2017,9,23.71875 })
    { 2017.,9.,23.8854167 }
sun_αδ({ 2017.,9.,23.8854167 })
    { "α:12h03m47s", "δ:0°24'32"" }
sun_ah(

```

```

1.1 1.2 *AstroLab_5 DEG
( 2017.,9.,23.8854167 )
touniversaltime({ 2017,9,23.71875 })
    { 2017.,9.,23.8854167 }
sun_αδ({ 2017.,9.,23.8854167 })
    { "α:12h03m47s", "δ:0°24'32"" }
sun_ah({ 2017.,9.,23.8854167 })
    { 252.950094476,15.4647870767 }

```

Il est pratique de transformer le résultat en DMS!

Le Soleil était donc un azimut de 252°57' (mesuré à partir de Nord) et à une altitude 15°27'.

```

1.1 1.2 *AstroLab_5 DEG
( 2017.,9.,23.8854167 )
sun_αδ({ 2017.,9.,23.8854167 })
    { "α:12h03m47s", "δ:0°24'32"" }
sun_ah({ 2017.,9.,23.8854167 })
    { 252.950094476,15.4647870767 }
( { 252.95009447564,15.464787076725 } ) ▶▶
{ 252°57'0.340112304",15°27'53.23347621" }

```

6.1.4 seasons(y, heure avancée)

Cette fonction renvoie l'heure et la date du début des saisons pour l'année y donnée. L'indicateur **heure avancée** vaut 0 ou 1. Peu importe le contenu de la variable **lieu**, si l'utilisateur entre 0, la fonction donnera le temps en **heure universelle** et en **heure normale**. Par contre, avec 1, la fonction donnera le temps en **heure universelle** et en **heure locale** (avec l'heure avancée).

Exemple

Calculez le moment du début des saisons pour l'année 2017.

La partie de droite du tableau contient l'heure locale de chaque événement. À partir de l'historique, il suffit de déplacer le curseur pour obtenir chacun de ces moments.

```

1.1 1.2 *AstroLab_5 DEG
seasons(2017,1)
"Season" "Universal Time"
"Spring" "Monday March 20, 2017. 10
"Summer" "Wednesday June 21, 2017. 0
"Autumn" "Friday September 22, 2017. 1
"Winter" "Thursday December 21, 2017.

```

```

1.1 1.2 *AstroLab_5 DEG
seasons(2017,1)
"Universal Time"
"Monday March 20, 2017. 10:28:41"
"Wednesday June 21, 2017. 04:24:18"
"Friday September 22, 2017. 20:01:56"
"Thursday December 21, 2017. 16:27:46"

```

```

1.1 1.2 *AstroLab_5 DEG
seasons(2017,1)
"LocalTime"
"Monday March 20, 2017. 06:28:41"
"Wednesday June 21, 2017. 00:24:18"
"Friday September 22, 2017. 16:01:56"
"Thursday December 21, 2017. 11:27:46"

```

Comme on le voit, l'automne 2017 est officiellement arrivée le 22 septembre 2017 à 16h01m56s HAE pour Montréal!

Vérifions les coordonnées du Soleil à cet instant.

Ce moment correspond à une heure locale qu'il faut transformer en temps universel pour ensuite pouvoir calculer les coordonnées du Soleil. Ci-contre, une copie d'écran de cette démarche.

```

1.1 1.2 *AstroLab_5 DEG
Thursday December 21, 2017. 11:27:46
makedateishort({ 2017,9,22,16,1,56 })
{ 2017,9,22.6680092593 }
touniversaltime({ 2017,9,22.668009259259 }
{ 2017.,9.,22.8346644 }
sun_αδ({ 2017.,9.,22.8346644 })
{ "α:12h00m00s", "δ:0°00'01" }

```

6.2 La Lune

Les fonctions reliées à la Lune sont

- *moon_ah*({y,m,d,d}) : Renvoie l'azimut et l'altitude de la Lune;
- *moon_αδ*({y,m,d,d}) : Renvoie l'ascension droite et la déclinaison de la Lune;
- *phases_moon*({y,m,d,d},heure avancée) : Renvoie la date et l'heure des phases de la Lune;
- *illu_frac_moon*({y,m,d,d}) : Renvoie le rapport de la partie illuminée de la Lune sur le disque lunaire complet et l'angle de rotation du croissant;
- *moon_illu_view*({y,m,d,d}) : Programme permettant l'affichage du croissant de Lune;
- *moon_coord*({y,m,d,d}) : Renvoie la longitude apparente, la latitude apparente, la distance Terre-Lune en km, l'ascension droite, la déclinaison et la parallaxe. Cette fonction est disponible avec la touche **var**.

6.2.1 moon_coord({y,m,d,d})

Cette fonction renvoie une liste contenant les coordonnées de la Lune pour la liste de date courte fournie en argument et **correspondant au temps universel (TU)**. La liste obtenue n'est pas transformée, elle contient des valeurs numériques qui peuvent ensuite être réutilisées par une autre fonction ou un autre programme.

Exemple

Quelles sont les coordonnées géocentriques de la Lune le 23 septembre 2017, à 13h30m00s HAE ?

- ① Création de la liste de date courte;
- ② Transformer la liste au temps universel;
- ③ Calculer la position de la Lune.

La liste obtenue contient : $\{\lambda^\circ, \beta^\circ, \Delta_{km}, \alpha_{heure}, \delta^\circ, \pi^\circ\}$

Pour utiliser la distance Terre-Lune en km dans un autre calcul, il suffit de rappeler la troisième valeur de la liste.



```
makedatelshort({ 2017,9,23,13,30,0 })
{ 2017,9,23.5625 }

touniversaltime({ 2017,9,23.5625 })
{ 2017.,9.,23.7291667 }

moon_coord({ 2017.,9.,23.7291667 })
{ 222.423222715,5.07380851167,397105.21 }
```

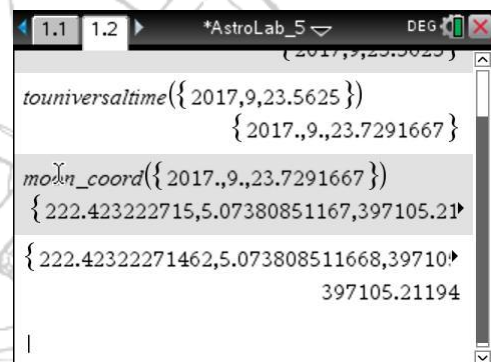


```
makedatelshort({ 2017,9,23,13,30,0 })
{ 2017,9,23.5625 }

touniversaltime({ 2017,9,23.5625 })
{ 2017.,9.,23.7291667 }

moon_coord({ 2017.,9.,23.7291667 })
{ 222.423222715,5.07380851167,397105.21 }

ans[3]
```



```
touniversaltime({ 2017,9,23.5625 })
{ 2017.,9.,23.7291667 }

moon_coord({ 2017.,9.,23.7291667 })
{ 222.423222715,5.07380851167,397105.21 }
{ 222.42322271462,5.073808511668,397105.21194 }
```

À ce moment, cette distance était de 397105,2 km.

6.2.2 moon_ad({y,m,d,d})

Pour obtenir plus rapidement l'ascension droite et la déclinaison de la Lune avec le format adéquat, il est préférable d'utiliser cette fonction.

Pour l'exemple précédent, **AstroLab_5** donne le résultat ci-contre.



```
moon_coord({ 2017.,9.,23.7291667 })
{ 222.423222715,5.07380851167,397105.21 }
{ 222.42322271462,5.073808511668,397105.21194 }

moon_ad({ 2017.,9.,23.7291667 })
{ "α:14h46m12s","δ:-10°43'31","Δ:397105.21194_km" }
```



```
moon_coord({ 2017.,9.,23.7291667 })
{ 222.423222715,5.07380851167,397105.21 }
{ 222.42322271462,5.073808511668,397105.21194 }

moon_ad({ 2017.,9.,23.7291667 })
{ "α:14h46m12s","δ:-10°43'31","Δ:397105.21194_km" }
```

À partir de l'historique des résultats, il suffit de se déplacer vers la droite pour voir tout le contenu de la liste. La distance Terre-Lune est cette fois fournie avec l'unité correctement affichée.

6.2.3 moon_ah({y,m,d,d})

Cette fonction renvoie l'azimuth et l'altitude de la Lune pour la liste de date courte donnée.
L'azimuth est mesurée à partir du Nord.

Exemple

Quel était la hauteur de la Lune le 23 septembre 2017 à Montréal, à 19h30m HAE?

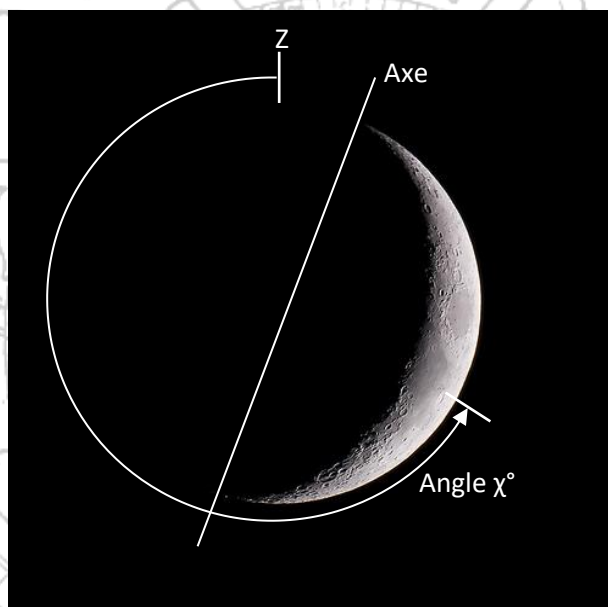
À ce moment, **AstroLab_5** calcule une hauteur de $13,1^\circ$.

```
1.1 1.2 *AstroLab_5 DEG
makedatelshort({ 2017,9,23,19,30,0 })
{ 2017,9,23.8125 }
touniversaltime({ 2017,9,23.8125 })
{ 2017.,9.,23.9791667 }
moon_ah({ 2017.,9.,23.9791667 })
{ 237.877289491,13.0718178336 }
```

6.2.4 illu_frac_moon({y,m,d,d})

Cette fonction renvoie deux nombre, k et χ .

- k : Rapport entre la partie illuminée et le disque lunaire complet;
- χ : Angle de rotation du croissant de la Lune comme le montre l'image suivante;



Exemple

Le 24 septembre 2017 à 19 heures HAE à Montréal, la fonction renvoie

$$k = 0,218$$

$$\chi = 279^\circ$$

À ce moment précis, le croissant représentait 21,8% de la Lune.

```
1.1 1.2 1.3 *AstroLab_5 DEG
makedatelshort({ 2017,9,24,19,0,0 })
{ 2017,9,24.79166666667 }
touniversaltime({ 2017,9,24.79166666667 })
{ 2017.,9.,24.9583333 }
illu_frac_moon({ 2017.,9.,24.9583333 })
{ 0.218084564575,279.215056253 }
```

6.2.5 moon_illu_view({y,m,d,d})

Ce programme prépare l’affichage de la Lune tel que vu au moment donné en argument.

Exemple

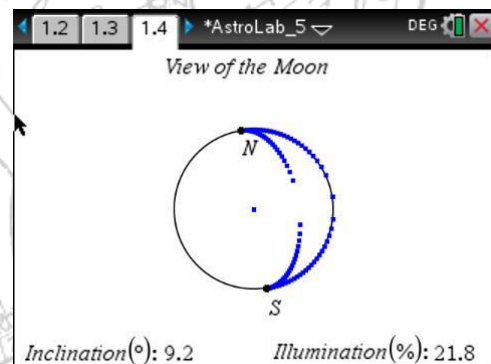
Quelle est la représentation du croissant de Lune pour l’exemple précédent, soit le 24 septembre 2017 à 19 heures HAE à Montréal ?

```
1.1 1.2 1.3 *AstroLab_5 DEG
makedatelshort({ 2017,9,24,19,0,0 })
{ 2017,9,24.79166666667 }
touniversaltime({ 2017,9,24.79166666667 })
{ 2017.,9.,24.9583333 }
illu_frac_moon({ 2017.,9.,24.9583333 })
{ 0.218084564575,279.215056253 }
moon_illu_view()
```

```
1.1 1.2 1.3 *AstroLab_5 DEG
{ 2017,9,24.79166666667 }
touniversaltime({ 2017,9,24.79166666667 })
{ 2017.,9.,24.9583333 }
illu_frac_moon({ 2017.,9.,24.9583333 })
{ 0.218084564575,279.215056253 }
moon_illu_view({ 2017.,9.,24.9583333 })
Done
```

Le croissant de Lune visible à ce moment précis est formé par l’ensemble des points bleus.

Puisque $\chi \approx 279^\circ$, l’axe Nord-Sud de la Lune subit une rotation d’environ 9° .



6.2.6 phases_moon({y,m,d,d},heure avancée)

Cette fonction renvoie l’heure et la date des phases de la Lune pour la liste de date courte fournie comme argument. L’indicateur **heure avancée** vaut 0 ou 1. Peu importe le contenu de la variable **lieu**, si l’usager entre 0, la fonction donnera le temps en **heure universelle et en heure normale**. Par contre, avec 1, la fonction donnera le temps en **heure universelle et en heure locale** (avec l’heure avancée).

Exemple

Quelle est l'heure locale pour Montréal, de la pleine Lune suivant le 25 septembre 2017?

```
1.1 1.2 1.3 *AstroLab_5 DEG
phases_moon({ 2017,9,25 },1)
"Phase of the Moon"      "Univer
"New Moon"      "Wednesday Septeml
"First Quarter"  "Thursday Septemb
"Full Moon"      "Thursday Octobe
"Last Quarter"   "Thursday Octobe
```

```
1.1 1.2 1.3 *AstroLab_5 DEG
phases_moon({ 2017,9,25 },1)
"Local Time"
Wednesday September 20, 2017. 01:29:57"
Wednesday September 27, 2017. 22:53:30"
Thursday October 05, 2017. 14:40:10"
Thursday October 12, 2017. 08:26:01"
```

AstroLab_5 indique qu'elle aura lieu le jeudi 5 octobre 2017 à 14h40m HAE.

6.3 Les planètes

Les fonctions reliées aux planètes sont

- *planet_ah(object,{y,m,d,d})* : Renvoie l'azimuth et l'altitude de la planète;
- *planet_αδ(object,{y,m,d,d})* : Renvoie l'ascension droite et la déclinaison de la planète;
- *planet_coord(object,{y,m,d,d})* : Renvoie la longitude apparente, la latitude apparente, la distance Terre-Planète en unités astronomiques, l'ascension droite et la déclinaison. Cette fonction est disponible avec la touche **var**.

6.3.1 planet_coord(object,{y,m,d,d})

Cette fonction renvoie une liste contenant les coordonnées géocentriques d'une planète pour la liste de date courte fournie en argument et **correspondant au temps universel (TU)**. La liste obtenue n'est pas transformée, elle contient des valeurs numériques qui peuvent ensuite être réutilisées par une autre fonction ou un autre programme.

Exemple

À quelle distance de la Terre se trouvait la planète Jupiter, le 26 septembre 2017 à 0h UT ?

```
1.1 1.2 1.3 *AstroLab_5 DEG
planet_coord("jupiter",{ 2017,9,26 })
{ 206.919992247,1.04696508607,6.3466927
```

```

1.1 1.2 1.3 *AstroLab_5 DEG
planet_coord("jupiter",{2017,9,26})
{206.919992247,1.04696508607,6.3466927}
ans[3]

```

```

1.1 1.2 1.3 *AstroLab_5 DEG
planet_coord("jupiter",{2017,9,26})
{206.919992247,1.04696508607,6.3466927}
{206.91999224739,1.0469650860687,6.34669276616}
|

```

Jupiter était à 6,35_au de la Terre.

6.3.2 planet_αδ(object,{y,m,d})

Cette fonction renvoie une liste dont le contenu est formaté.

Exemple

Quelle était l'ascension droite de Jupiter le 26 septembre 2017 à 0h UT?

AstroLab_5 donne 13h41m27s.

```

1.1 1.2 1.3 *AstroLab_5 DEG
planet_coord("jupiter",{2017,9,26})
{206.919992247,1.04696508607,6.3466927}
{206.91999224739,1.0469650860687,6.34669276616}
planet_αδ("jupiter",{2017,9,26})
{"α:13h41m27s","δ:-9°23'48","Δ:6.34669"}
|

```

6.3.3 planet_ah(object,{y,m,d})

Cette fonction renvoie l'azimut et l'altitude d'une planète, à un moment précis.

Exemple

Quelle était l'altitude de Saturne à Montréal, le 26 septembre 2017 à 20h HAE ?

Saturne était à une hauteur de 16,6°.

```

1.1 1.2 1.3 *AstroLab_5 DEG
{0.134127,0.92348,-6.34669}
makedatelshort({2017,9,26,20,0,0})
{2017,9,26.8333333333}
jouniversaltime({2017,9,26.8333333333})
{2017,9,26.9999884}
planet_ah("saturne",{2017,9,26.9999884})
{209.876318675,16.6419789169}
|

```

6.3.4 Le nom des planètes

AstorLab_5 accepte les noms suivants.

Planète	Noms acceptés par <i>AstroLab_5</i>
<i> Mercure </i>	mer, mercure, mercury
<i> Vénus </i>	ven, venus
<i> Mars </i>	mar, mars
<i> Jupiter </i>	jup, jupiter
<i> Saturne </i>	sat, saturn, saturne
<i> Uranus </i>	ura, uranus
<i> Neptune </i>	nep, neptune

6.3.5 Les coordonnées héliocentriques d'une planète

La fonction `co_helio_eclip(Object, {y, m, d. d})` renvoie les coordonnées héliocentriques

$\{L: \text{longitude}, B: \text{Latitude}, R: \text{Distance au Soleil (au)}\}$ d'une planète à un moment précis donné par la liste de date courte $\{y, m, d. d\}$. Cette fonction est disponible avec la touche **var**.

Exemple

Calculez les coordonnées héliocentriques de Vénus le 20 décembre 1992 à 0h TD. Pour cet exemple, $\Delta T = 0$!



```
1.1 1.2 1.3 *AstroLab_5 DEG
initialisation() Done
co_helio_eclip("ven", {1992, 12, 20})
{26.1143023658, -2.62092117779, 0.724603}
```

6.4 Pluton

AstroLab_5 comporte trois fonctions permettant de calculer les coordonnées de Pluton.

- `pluto_ah({y, m, d. d})` : Renvoie l'azimuth et l'altitude de Pluton;
- `pluto_αδ({y, m, d. d})` : Renvoie l'ascension droite et la déclinaison de Pluton;
- `pluto_coord({y, m, d. d})` : Renvoie l'ascension droite, la déclinaison, la distance Terre-Pluton en *unités astronomiques* et la distance Soleil-Pluton aussi en *unités astronomiques*. Cette fonction est disponible avec la touche **var**.

6.4.1 pluto_coord({y,m,d.d})

Cette fonction renvoie la liste $\{\alpha(hr), \delta^\circ, \Delta(au), r(au)\}$ pour la liste de date courte donnée, correspondant au temps universel.

Exemple

Quelle était la distance Soleil-Pluton le 26 septembre 2017 à 0h TU?

```
lieu { 45.61,73.57,-5,1,68 }
pluto_coord({ 2017,9,26 })
{ 19.1951855175,-21.7857514087,33.16146
ans[4]
```

Après avoir exécuté la fonction, il suffit de rappeler le quatrième élément de la liste pour avoir une distance d'environ 33,4 unités astronomiques.

```
lieu { 45.61,73.57,-5,1,68 }
pluto_coord({ 2017,9,26 })
{ 19.1951855175,-21.7857514087,33.16146
{ 19.195185517511,-21.785751408651,33.1
33.4139238188
|
```

6.4.2 pluto_αδ({y,m,d.d})

Cette fonction renvoie la liste formatée $\{\alpha(HMS), \delta^\circ, \Delta(au)\}$.

Exemple

Pour l'exemple précédent, nous obtenons :

```
lieu { 45.61,73.57,-5,1,68 }
pluto_coord({ 2017,9,26 })
{ 19.1951855175,-21.7857514087,33.16146
{ 19.195185517511,-21.785751408651,33.1
33.4139238188
pluto_αδ({ 2017,9,26 })
{ "α:19h11m43s","δ:-21°47'09","Δ:33.161
|
```

6.4.3 pluto_ah({y,m,d.d})

Cette fonction renvoie l'azimut et l'altitude de Pluton.

Exemple

À Montréal, le 26 septembre 2017 à 0h UT, Pluton était à un azimuth de 183°30' et à une hauteur de 22°31'.

```
lieu { 45.61,73.57,-5,1,68 }
pluto_αδ({ 2017,9,26 })
{ "α:19h11m43s","δ:-21°47'09","Δ:33.161
pluto_ah({ 2017,9,26 })
{ 183.509328707,22.5294679632 }
({ 183.5093287069,22.529467963189 })▶D▶
{ 183°30'33.58334484",22°31'46.084667480
|
```


6.5 Les satellites de Jupiter

Le programme `sat_jup_view({y, m, d. d})` permet d'afficher une vue des quatre plus gros satellites de Jupiter tel que vue de la Terre pour une liste de date courte correspondant à un moment précis.

Exemple

Quelle était la position des quatres plus gros satellites de Jupiter le 27 septembre 2017 à 23h30m00s HAE?

```

1.1 1.2 1.3 *AstroLab_5 DEG
makedatelshort({ 2017,9,27,23,30,0 })
{ 2017,9,27.9791666667 }
touniversaltime({ 2017,9,27.979166666666 })
{ 2017.,9.,28.1458218 }
sat_jup_view()

```

Une fois la liste de date courte correspondant au temps universel de ce moment obtenue, la commande

`sat_jup_view({2017,9,28.1458218})`

renvoie un tableau donnant les coordonnées cartésiennes des satellites **Io**, **Europa**, **Ganymède** et **Callisto**.

La première ligne du tableau correspond aux coordonnées de **Io**.

Dans un plan cartésien dont l'unité correspond au rayon de la planète Jupiter, nous obtenons la position réelle des satellites, telle que vu de la Terre.

```

1.1 1.2 1.3 *AstroLab_5 DEG
{ 2017.,9.,28.1458218 }
sat_jup_view({ 2017.,9.,28.1458218 })
[ -1.98273939249 -0.264582493863
  7.51724036145  0.262903014469
  14.2347623012  0.227249259206
  15.477186692  -1.02797704318 ]
Done

```

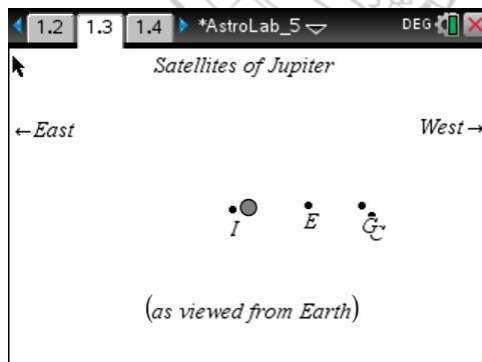
La vue de la Terre des satellites de Jupiter.

Ici, nous avons de gauche à droite :

Io, Jupiter, Europa, Ganymède et Callisto

La vue à travers une lunette astronomique.

Il suffit dans ce cas d'inverser la *Ti-Nspire*!



6.6 Les anneaux de Saturne

La fonction `saturn_ring({y,m,d})`, renvoie la liste $\{a'', b'', B^\circ, p^\circ, r_{ua}\}$, où

- a'' : La mesure du grand axe de l'ellipse extérieure de l'anneau externe en secondes d'arc;
- b'' : La mesure du petit axe de l'ellipse extérieure de l'anneau externe en secondes d'arc;
- B° : La mesure de l'angle d'inclinaison (en degrés) des anneaux tel que vu de la Terre;
- p° : La mesure de l'angle d'inclinaison des anneaux par rapport au Nord céleste;
- r_{ua} : La distance Terre-Saturne en unités astronomiques.

Pour obtenir les dimensions des autres anneaux principaux de Saturne, il faut multiplier les paramètres a'' et b'' par :

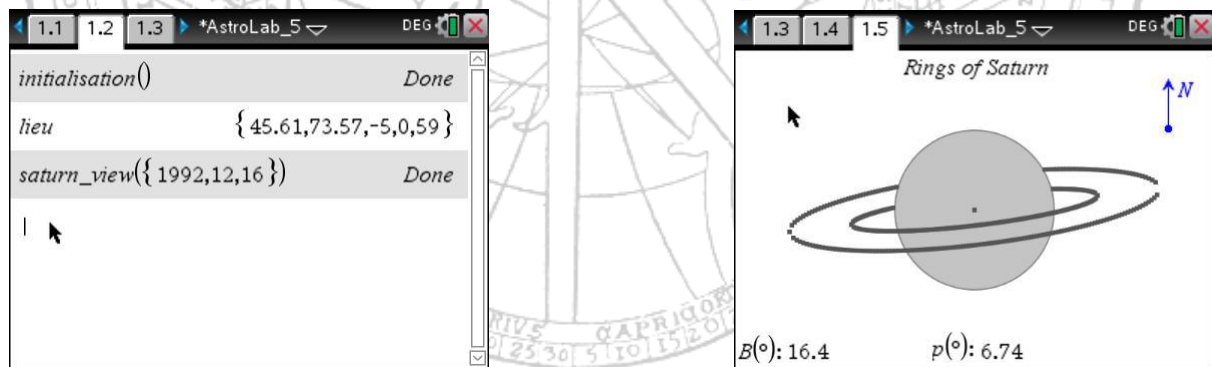
- 0.8801 : pour obtenir les dimensions de l'ellipse intérieure de l'anneau extérieur;
- 0.8599 : pour obtenir les dimensions de l'ellipse extérieure de l'anneau interne;
- 0.6650 : pour obtenir les dimensions de l'ellipse intérieure de l'anneau interne;
- 0.5486 : pour obtenir les dimensions de l'ellipse interne de l'anneau de poussières.

Le programme `saturn_view({y,m,d})` utilise la fonction précédente pour ensuite préparer l'affichage du schéma de Saturne pour le moment fournit en argument.

Exemple

Quelle était l'aspect des anneaux de Saturne le 16 décembre 1992?

En 1992, $\Delta T = 59$ secondes.



Attention

Il faut réinitialiser : $\Delta T = 68$ secondes pour l'année 2018.

6.7 Lever et coucher des astres

Le programme `rising_tr_set(object, {y, m, d}, time)`, renvoie l'heure du lever, du transit et du coucher pour les planètes, Pluton, la Lune et le Soleil.

L'indicateur **time** contrôle le temps au moment de l'affichage des résultats.

Si

- *temps* = 1, alors la sortie affiche un temps en *heures normales*;
- *temps* = 2, alors la sortie affiche un temps en *heures locales*;
- Pour les autres valeurs, la sortie affiche un temps en *heures universelles*.

Attention,

Ce programme utilise le contenu de la variable *lieu*.

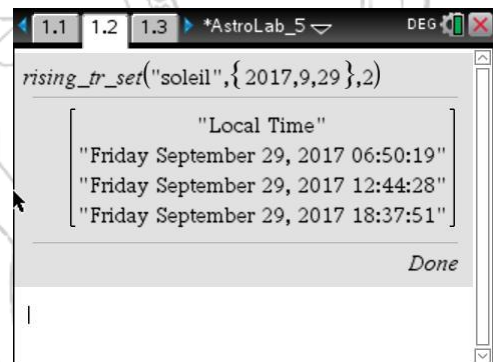
En plus des noms des planètes (voir section 6.3.4), le programme accepte les noms suivants :

Objets	Noms acceptés par <i>AstroLab_5</i>
<i>Soleil</i>	sol, sun, soleil
<i>Lune</i>	lune, moon

Exemple

Quelle est l'heure locale du coucher du Soleil pour Montréal, le 29 septembre 2017?

AstroLab_5 indique 18h37m51s HAE.



```
1.1 1.2 1.3 *AstroLab_5 DEG
rising_tr_set("soleil",{2017,9,29},2)

    "Local Time"
    "Friday September 29, 2017 06:50:19"
    "Friday September 29, 2017 12:44:28"
    "Friday September 29, 2017 18:37:51"

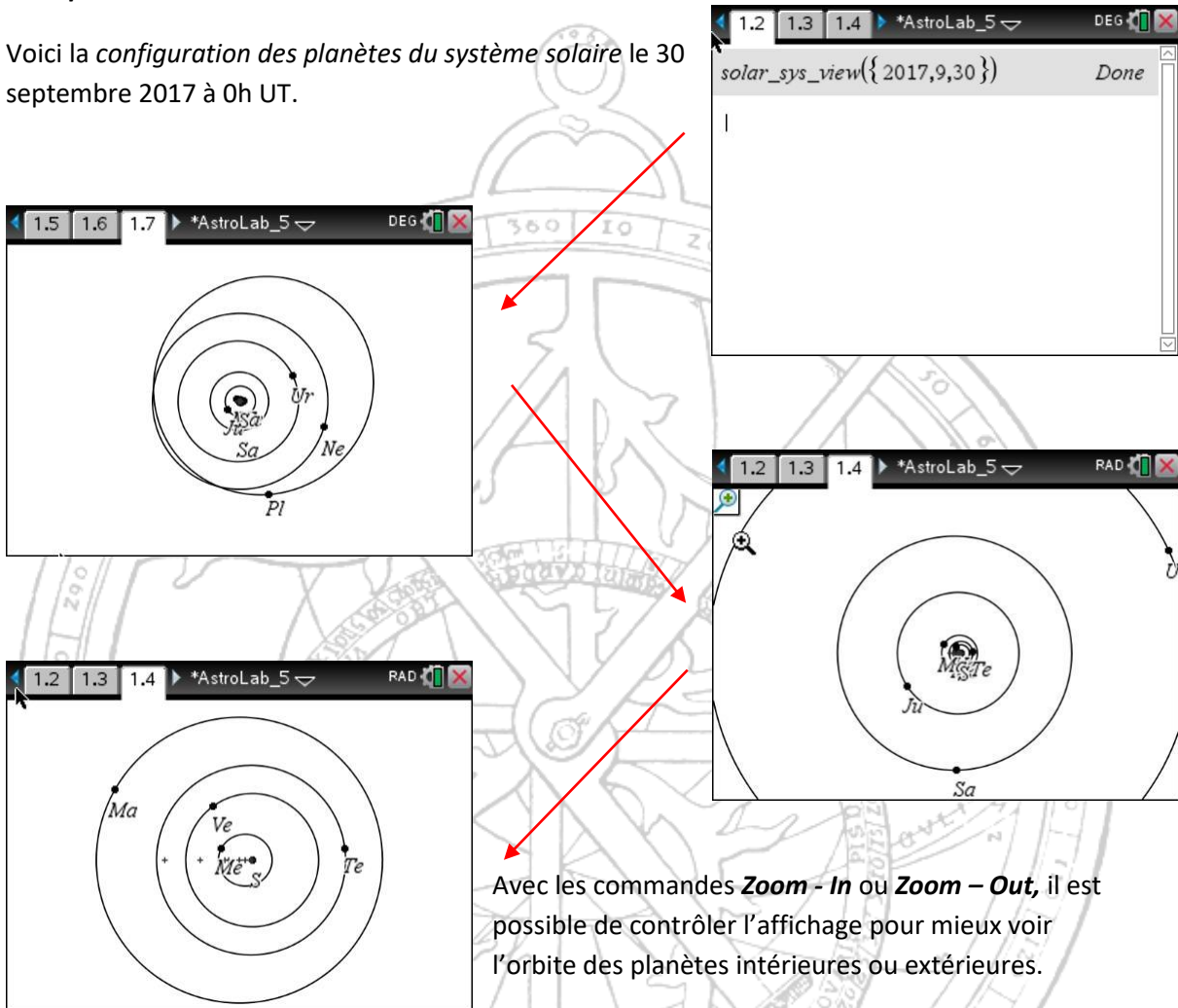
    Done
```

6.8 La configuration du système solaire

Le programme `solar_sys_view({y, m, d. d})` permet l'affichage graphique de la configuration des planètes du système solaire.

Exemple

Voici la *configuration des planètes du système solaire* le 30 septembre 2017 à 0h UT.



6.9 Carte du ciel

Le programme `starschart_view({y,m,d,d})` permet d'obtenir pour l'instant donné en argument, une carte du ciel pouvant montrer jusqu'à 1 469 étoiles ayant une magnitude allant de 1 à 6.

Exemple

Quelle était la position des planètes le 13 mars 2018 à Montréal à 0h TU?

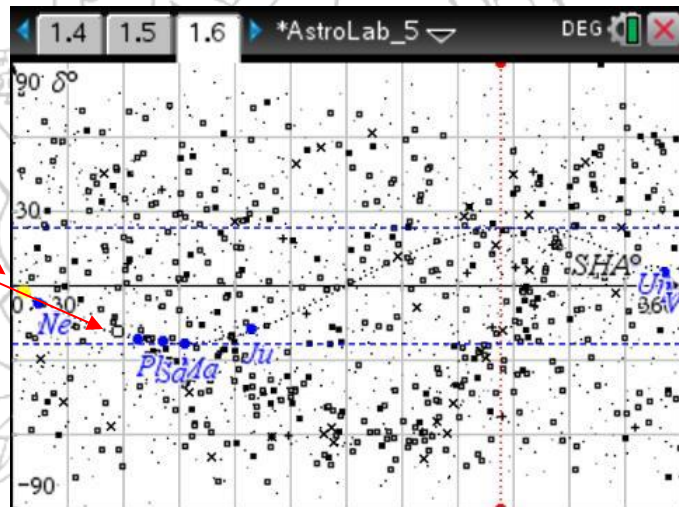
Ce programme affiche quelques informations concernant les calculs en cours durant son exécution.

Une fois terminé, le sixième onglet permet de voir la carte du ciel de cet instant!



Sur cette carte,

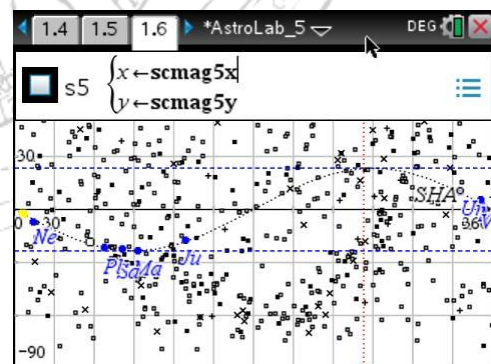
- **Le point jaune** représente le Soleil;
- **Le point blanc** représente la Lune;
- **Les points bleus**, les planètes;
- **Les traits pointillés bleus**, les tropiques;
- **La sinusoïde**, l'écliptique;
- **Le trait vertical rouge en pointillés**, le méridien local d'observation du moment fournit comme argument.

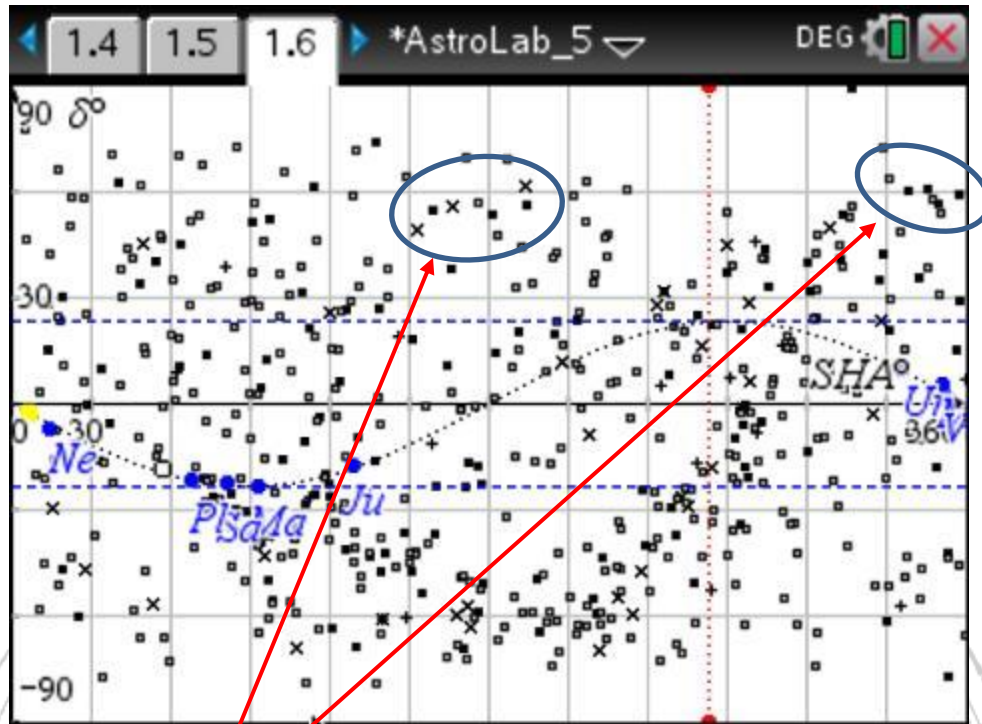


Il est possible de modifier le nombre d'étoiles sur la carte en choisissant le ou les nuages de données que la calculatrice affichera. Le numéro du nuage indique la magnitude des étoiles.

En décochant les nuages 5 et 6, seules les étoiles de magnitude 1 à 4 seront affichées.

L'image de la page suivante montre la carte du même instant avec maintenant moins d'étoiles.





- La Grande Ourse
- Cassiopée

Quelques informations utiles concernant les nuages de points.

Nom du nuage	Magnitude	Nombre d'étoiles	Symbole
scmag1x scmag1y	$M \leq 1$	16	+
scmag2x scmag2y	$1 < M \leq 2$	35	×
scmag3x scmag3y	$2 < M \leq 3$	124	▪
scmag4x scmag4y	$3 < M \leq 4$	341	◻
scmag5x scmag5y	$4 < M \leq 5$	635	.
scmag6x scmag6y	$M > 5$	318	.

7. Derniers mots

AstroLab_5 est un outil de calculs permettant de mieux comprendre l'astronomie ou la mécanique céleste. Utilisé avec soin, il donnera des résultats de très grandes précisions. Pour mieux le comprendre, je vous conseille l'observation directe d'un astre à l'œil nu avec une montre bien ajustée et une simple boussole. Finalement, je permets à quiconque qui le désire, de transformer et, je l'espère, d'améliorer **AstroLab_5**.

Marcel Pelletier (Octobre 2017)

