

Que faire avec un tableur formel
au collège et au lycée ?
Applications avec CmathOOoCAS

Christophe.Devalland@ac-rouen.fr

Table des matières

I.Présentation de CmathOOoCAS.....	2
1.OOoCalc travaille maintenant dans C.....	2
2.OOoCalc sait manipuler des expressions mathématiques.....	3
3.Un éditeur de fonctions permet d'étendre à l'infini les capacités du tableur.....	4
4.... et de travailler sur des algorithmes.....	6
II.Présentation des activités.....	6
III.Fiches de travail.....	8
1.Prendre en main CmathOOoCAS.....	8
2.Des activités pour les élèves.....	9
3.Des problèmes dont l'activité est à créer.....	13

I. Présentation de CmathOOoCAS

CmathOOoCAS est une extension libre et gratuite écrite en C++ pour le module Calc de OpenOffice.org (noté ensuite OOoCalc). Elle s'appuie sur la librairie de calcul formel GIAC de XCAS (CAS est l'acronyme de « Computer Algebra System », que l'on peut traduire par « système de calcul formel ». La définition qu'en donne Wikipédia est : « Un système de calcul formel est un logiciel qui facilite le calcul symbolique. La partie principale de ce système est la manipulation des expressions mathématiques sous leur forme symbolique. »).

Le but de CmathOOoCAS est donc d'étendre les capacités de OOoCalc afin de rendre le tableur plus ouvert aux objets mathématiques que nous voulons manipuler.

En effet, les fonctions mathématiques incluses dans OOoCalc ne savent travailler que sur des nombres décimaux et, comme les calculatrices courantes, les limites du logiciel sont vite atteintes. Impossible par exemple de travailler sur le nombre réel $\sqrt{2}$ ou sur une expression polynomiale telle que $x^2 - x + 1$. Alors qu'aujourd'hui, certaines calculatrices utilisées au collège sont capables de calculer avec des radicaux, il est frustrant que le tableur hébergé par un ordinateur n'en soit pas capable ! Ainsi est venue l'idée d'intégrer un moteur de calcul formel dans le tableur d'OpenOffice.org. Ce projet n'aurait d'ailleurs certainement pas été possible avec un autre tableur. Il fallait une architecture ouverte et documentée pour venir y greffer mon programme. OpenOffice.org présentait les qualités requises.

J'en profite ici pour remercier vivement ceux qui m'ont aidé dans l'élaboration de mon programme et sans qui CmathOOoCAS n'existerait pas :

- Serge Moutou, auteur d'une remarquable documentation sur la programmation d'une extension en C++ pour OpenOffice.org. Il m'a permis de croire que ce projet était réalisable.
- Bernard Parisse, auteur de la librairie GIAC, moteur de XCAS et maintenant de CmathOOoCAS. Sa disponibilité et sa gentillesse ont aidé grandement à la réalisation de mon projet.
- Les développeurs de chez Sun qui m'ont ponctuellement apporté des précisions techniques indispensables.

Voyons maintenant les possibilités de CmathOOoCAS. Les apports peuvent se résumer en quatre parties :

1. OOoCalc travaille maintenant dans C


Un des avantages du tableur est de pouvoir enchaîner les calculs, chaque résultat contenu dans une cellule servant de base au calcul de la cellule suivante. La précision du calcul est en général suffisante pour nos besoins et pour éviter des résultats trop faux à cause des erreurs d'arrondis. Cependant, on ne peut pas être certain que ce qui est observé, par exemple en étudiant une suite, n'est pas faussé par les limitations en calculs numérique du logiciel. Certaines suites convergentes divergent parfois à cause d'accumulations d'erreurs d'arrondis. Pouvoir travailler avec des valeurs exactes, quitte à évaluer le résultat si besoin, est plutôt rassurant.

Prenons comme exemple la suite qui approche $\sqrt{2}$ par la méthode de Héron (u_0 étant fixé, on définit cette suite récurrente par $u_{n+1} = \frac{1}{2}(u_n + \frac{2}{u_n})$). Le calcul des premiers termes dans le tableur-formel nous donne une suite de rationnels convergeant vers l'irrationnel $\sqrt{2}$. Les calculs seront exacts, il n'y a aura aucun risque d'erreurs d'arrondis. Cependant, il est plus parlant d'observer la convergence de cette suite à partir des valeurs approchées. Utilisons donc la fonction « `evaluer(nombre;nb_décimales)` » permettant d'obtenir une valeur approchée avec autant de décimales que l'on souhaite. Ici avec 40 décimales :

Là aussi, voir toutes ces fractions n'est pas très parlant. Même s'il est satisfaisant pour l'esprit d'obtenir une suite de valeurs exactes, il sera plus utile de demander des valeurs approchées. Toutefois, cela permet de remarquer que cette suite est la même que celle obtenue avec la méthode de Héron. D'ailleurs, le calcul de x_{n+1} en fonction de x_n permet de le justifier rapidement.

Pour aller plus loin, on pourra construire un algorithme qui calculera par dichotomie des valeurs approchées de $\sqrt{2}$ et vérifier que c'est la méthode de Newton converge le plus vite...

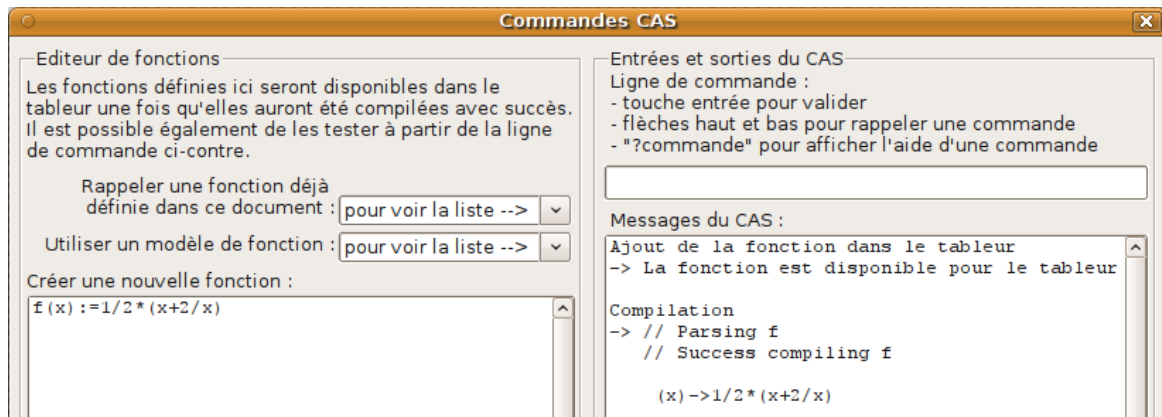
3. Un éditeur de fonctions permet d'étendre à l'infini les capacités du tableur...

Une boîte de dialogue contenant un éditeur de fonctions ainsi qu'une ligne de commandes de calculs formels est accessible via le menu CmathOOoCAS ou grâce à l'icône  de la barre de menu CmathOOoCAS. La ligne de commande est disponible dans la partie droite de cette boîte pour y taper des instructions comme dans Xcas et la partie gauche est composée d'un éditeur permettant de créer ses propres fonctions. Plusieurs langages de programmation sont utilisables dans CmathOOoCAS :

- Xcas
- Maple
- TI89
- Mupad
- « algorithmique » (proche du langage naturel).

Une fois définie et compilée sans erreur, la fonction devient disponible dans le tableur comme n'importe quelle autre fonction.

Reprenons la suite de Héron. Il est possible de définir une fonction $f(x) = \frac{1}{2}(x + \frac{2}{x})$ puis de la compiler :

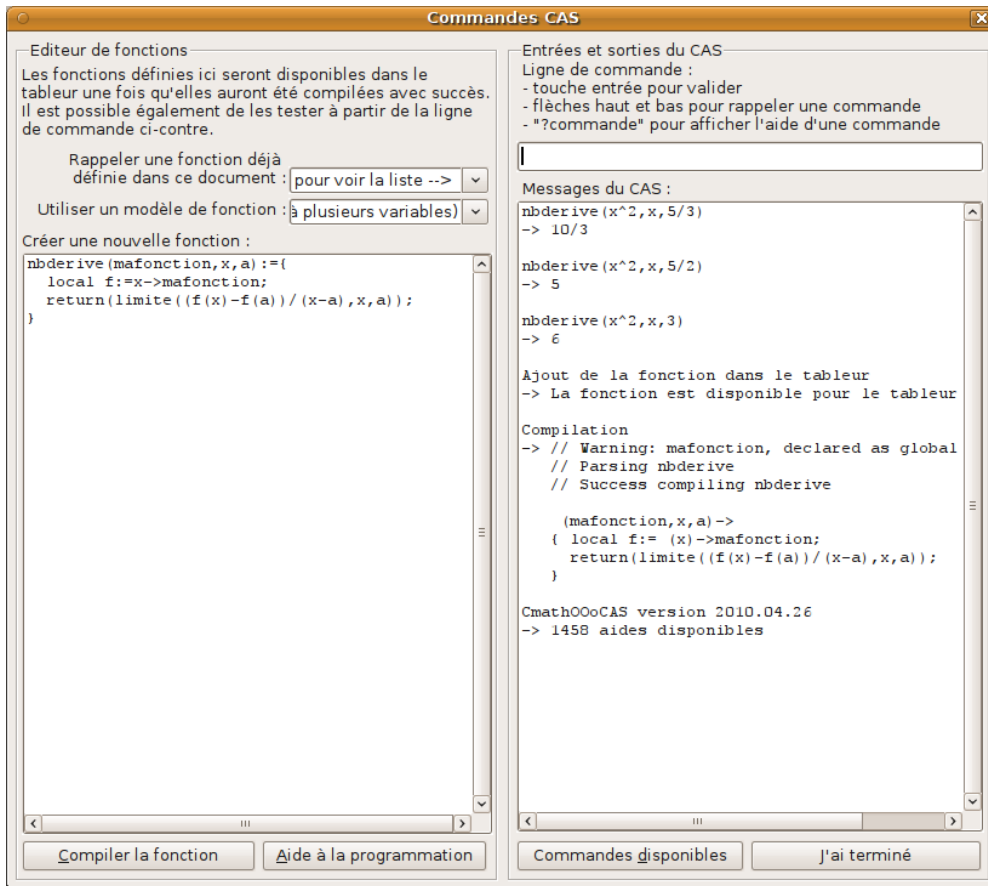


Pour calculer les premiers termes de la suite, il suffira donc d'entrer la formule $=f(B2)$ et de recopier cette formule vers le bas.

Prenons un autre exemple : je souhaite dans mon cours faire le lien entre les notions de nombre dérivé et de fonction dérivée. Je vais créer une nouvelle fonction nommée « nbderive » qui me renverra la limite du taux d'accroissement en a . Avec la syntaxe Xcas, cela donne :

```
nbderive(mafonction, x, a) := {
  local f:=x->mafonction;
  return(limite((f(x)-f(a))/(x-a), x, a));
}
```

On peut la tester avec la ligne de commande. Voilà ce que l'on obtient :



et son utilisation dans le tableur en tant que nouvelle fonction :

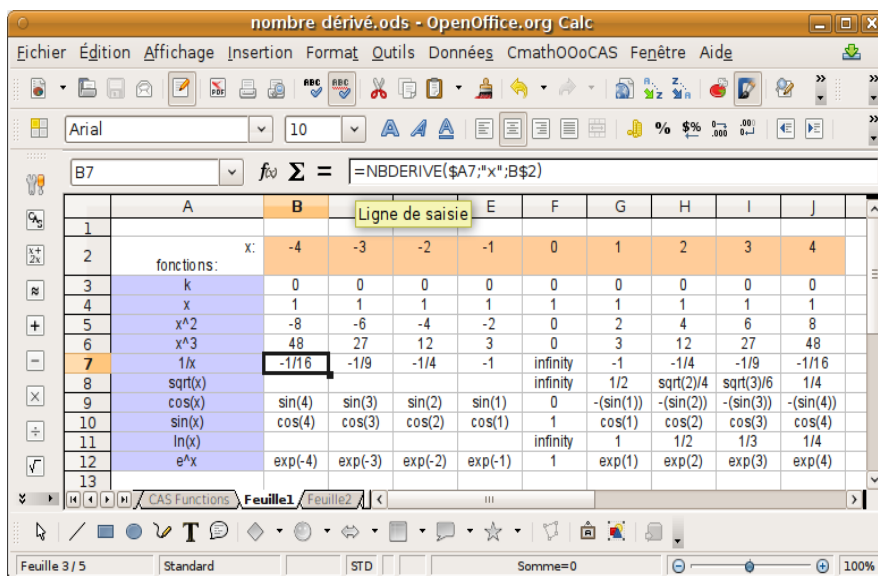


Illustration 3: du fait que les résultats renvoyés par la fonction sont exacts, il est possible de définir et conjecturer les fonctions dérivées des fonctions usuelles. Pour autant, l'usage de l'outil informatique n'empêche pas de réfléchir : on pourra, par exemple, demander d' cliquer l'apparition d'un signe moins sur la ligne de $\cos x$ et la présence des « infinity » dans la colonne du 0. Avec un tableur numérique, il ne serait pas raisonnable de faire retrouver les expressions de toutes ces fonctions dérivées à partir des valeurs approchées..

Suite et fin de cet exemple dans la partie II...

4. ... et de travailler sur des algorithmes.

La présence de plusieurs syntaxes, dont « algorithmique » et « TI », permet de s'initier sans trop d'efforts et de faire tourner ses algorithmes dans l'environnement familier du tableur. On bénéficiera, en plus, de l'avantage du calcul formel et des valeurs exactes. Quoi de plus énervant en effet de tester son algorithme de détection d'un triangle rectangle et de voir qu'il ne sert à rien puisqu'en lui donnant les longueurs 1, $\sqrt{2}$ et $\sqrt{3}$ il s'obstinera à répondre : « le triangle n'est pas rectangle » (sauf à tricher sur les arrondis et à introduire de faux-positifs !).

II. Présentation des activités

Les activités proposées au paragraphe suivant sont réparties sur trois niveaux, selon les besoins ou les envies du lecteur :

1. des énoncés très guidés, pas à pas, pour découvrir les fonctionnalités de CmathOOoCAS.
2. des activités telles que je les donne à mes élèves.
3. des problèmes mathématiques qu'il serait intéressant de résoudre avec un tableur formel, dont l'activité reste à rédiger.

Dans les trois cas, l'idée de départ est la même : partir d'un problème mathématique qui sera exploré, voire résolu, avec l'outil informatique. C'est l'assurance, de mon point de vue, de ne pas construire une activité artificielle, trop dépendante d'un logiciel. C'est l'outil qui doit se mettre au service des mathématiques et non l'inverse.

En gardant cette idée de base en tête, on évite de tomber dans le piège d'un énoncé contenant des tas de consignes techniques telles que : « inscrire la formule x dans la cellule y » ou « aller dans le menu x chercher l'instruction y ». Le problème mathématique posé doit amener l'élève à faire preuve d'autonomie afin de choisir la bonne commande de l'outil adéquat. A cet égard, on pourra prendre exemple sur les énoncés proposés pour l'expérimentation de l'épreuve pratique de Terminale S. Les textes n'évoquent pratiquement jamais l'aspect technique d'un outil particulier et toutes les questions sont d'ordre mathématique. D'ailleurs, le choix de tel ou tel logiciel n'est pratiquement jamais évoqué dans ces énoncés ; choisir un outil (tableur, logiciel de géométrie dynamique, grapheur, ...) est souvent la première question implicite de ces sujets. Une fois le logiciel choisi, sélectionner la bonne commande fait aussi partie de l'exercice et demande de l'autonomie. Je pense que, sauf cas particuliers, il n'est pas judicieux d'orienter les élèves sur ce point. Je suis toujours agréablement surpris de voir à quel point on met en évidence des lacunes des élèves concernant la nature des objets mathématiques à manipuler pendant une activité. Avec géoplan par exemple, qui est à mon avis inégalable pour obliger les élèves à comprendre ce qu'ils sont en train de construire, tel élève ne comprend pas qu'il ne voie pas apparaître dans son repère la variable numérique qu'il vient de créer, tel autre hésite devant les menus « créer un point libre dans le plan » et « créer un point repéré ». Sans avoir bien réfléchi à ce que l'on veut construire et aux objets que l'on doit manipuler, le logiciel ne sera pas d'un grand secours. Ainsi, contrairement à ce que l'on pourrait penser au premier abord, moins l'énoncé comporte de consigne technique et plus le questionnement des élèves et les réponses du professeurs pendant la séance seront de nature mathématique (« quelle différence faites-vous entre une variable numérique et les objets affichés dans ce repère ? » « l'énoncé donne-t-il un renseignement sur le point M ? est-il fixé ou quelconque ? »).

Pour ma part, j'ai essayé également de susciter cette réflexion dans CmathOOoCAS en organisant les fonctions disponibles en catégories (« fonctions usuelles », « numérique », « algèbre », « équations », etc...), obligeant le lecteur à bien maîtriser la nature de l'objet qu'il veut manipuler (cf. la rubrique « les fonctions » du menu CmathOOoCAS dans le tableur).

La deuxième grande idée qui guide la conception de mes activités est l'articulation entre « expérimentation » et « conjecture ». C'est bien sûr la qualité première de l'ordinateur : calculer rapidement pour explorer de nombreuses situations. La plupart des activités utilisant le tableur respectent cette logique. Avec CmathOOoCAS, il est possible d'aller plus loin. L'association de la feuille de calcul et du calcul formel permet souvent une étude plus profonde des phénomènes étudiés car l'apport du calcul formel est loin d'être négligeable : il permet de changer la nature de l'expérimentation. Là où le "tableur classique" se limite à une exploration numérique, le "tableur formel" permet d'achever le travail par une réelle validation de la conjecture émise pendant l'activité en s'affranchissant des valeurs numériques. Ne disions nous pas à nos élèves que le résultat fourni par une machine ne pouvait pas servir de preuve mathématique ? Les activités présentées plus bas prouvent le contraire... même si une validation uniquement informatique peut être troublante pour l'esprit (c'est pourtant une démarche couramment acceptée de nos jours, depuis la démonstration du célèbre théorème des quatre couleurs qui partagea la communauté scientifique en 1976).

Reprenons l'exemple d'utilisation de la fonction « `nbderive` ». L'ajout d'une simple colonne et la recopie des formules permet de confirmer ou d'infirmer les expressions supposées des fonctions dérivées :

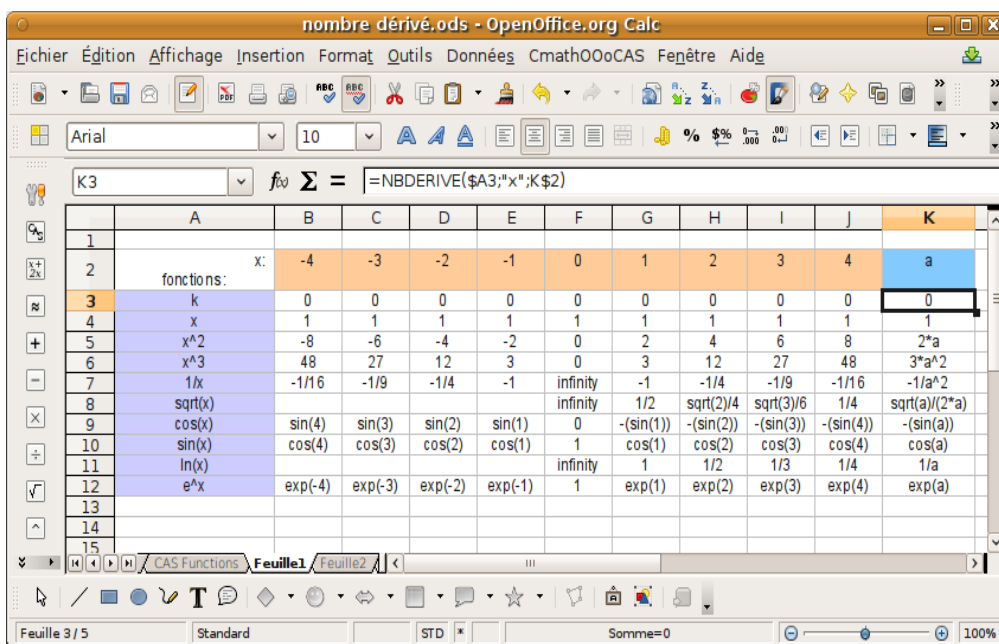


Illustration 4: on pourrait penser qu'il n'y a plus rien à faire, tellement les résultats s'obtiennent facilement ! Je ne le pense pas. Nous avons vu précédemment que l'utilisation du calcul formel permettait aussi de travailler sur les propriétés de certaines fonctions (que se passe-t-il en 0 lorsqu'on lit « infinity » ? Pourquoi ne voit-on pas $-\sin(\dots)$ dans toutes les colonnes ?). Rien n'empêche ensuite de reprendre le crayon pour faire retrouver à la main, par le calcul, l'une des expressions de la colonne K ou bien pour transformer l'expression de la fonction dérivée de $x \rightarrow \sqrt{x}$ qui n'apparaît pas ici dans sa forme usuelle...

Le troisième point commun des activités présentées ici est qu'elles tirent avantage, soit du calcul exact, soit du calcul formel. Elles ne seraient pas réalisables intégralement sans le couple "tableur-calcul formel".

Les fiches de travail qui suivent s'apprécieront en se mettant dans la peau d'un élève, devant une feuille de calcul vierge... Bonne découverte !

III. Fiches de travail

1. Prendre en main CmathOOoCAS

a) le carré d'un nombre se terminant par 5

Une méthode simple permet de calculer mentalement le carré d'un nombre N se terminant par 5. Essayons de découvrir cette méthode puis de la justifier.

1. A l'aide d'une feuille de calcul, nous allons balayer les nombres N de 15 à 155 (ou davantage selon vos envies...) Dans la première colonne, on placera le nombre de dizaines contenues dans N . Dans la seconde colonne, placer le chiffre des unités de N (donc 5). Dans la troisième colonne, calculer la valeur de N .
2. Calculer dans la quatrième colonne le carré de N . Dans la cinquième colonne, calculer le produit $d \times (d + 1)$ où d désigne le nombre de dizaines contenues dans N .
3. En observant les résultats obtenus, expliquer comment il semble possible de calculer mentalement le carré d'un nombre se terminant par 5.

Validation de la conjecture : nous allons suivre la même démarche que précédemment mais travailler sur un nombre N qui soit quelconque.

4. Dans la colonne contenant les nombres de dizaines, remplacer 15 par d (entrer " d ") Les colonnes suivantes devraient afficher #VALEUR!
Ceci est dû au fait que les opérations mathématiques de base du tableur (+, -, *, /, ^) ne savent travailler que sur des valeurs numériques alors que nous voulons effectuer des calculs littéraux.
Les fonctions ajoutées par CmathOOoCAS pouvant opérer aussi bien sur des valeurs numériques que sur des expressions formelles (entrées sous forme de texte, entre guillemets), les opérations de base pourront donc être

remplacées par les fonctions CSOMME(), CSOUSTRACTION(), CPRODUIT(), CQUOTIENT() et CPOUISSANCE(). Des raccourcis sont disponibles sous forme d'icônes dans la barre de menu.

5. Modifiez les formules de la feuille de calcul pour n'utiliser que les fonctions de CmathOOoCAS. Les "#VALEUR!" devraient disparaître.
6. Démontrer alors la conjecture émise à la question 3.

b) Comparaison de deux fractions

Des règles vues au collège permettent, dans certains cas favorables, de comparer deux fractions. Voici un problème où ces règles ne sont pas applicables :

On veut comparer les fractions $\frac{n}{n+1}$ et $\frac{n+1}{n+2}$ pour $n \in \mathbb{N}$.

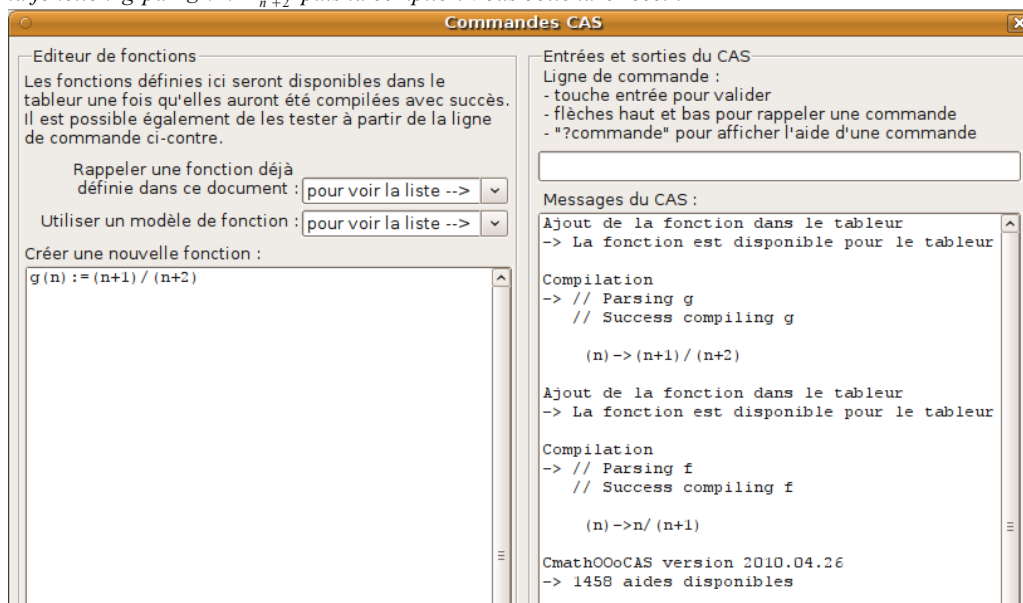
1. Construire un tableau faisant apparaître les valeurs exactes et arrondies à 10^{-6} près de ces deux fractions pour $n \leq 15$.

aide : plusieurs constructions sont possibles.

a) construire une colonne « n » contenant les entiers de 0 à 15 puis à sa droite une colonne « n+1 » et une colonne « n+2 ». On obtient les valeurs exactes des fractions demandées en créant deux nouvelles colonnes à partir des formules =cquotient(A1;B1) et =cquotient(B1;C1)

b) construire une colonne « n » contenant les entiers de 0 à 15 puis à sa droite une colonne à partir de la formule =substituer("n/(n+1)";"n";A1) et une autre avec =substituer("(n+1)/(n+2)";"n";A1)

c) A l'aide de l'éditeur de fonctions (icône CAS), définir une fonction f par $f(n) = \frac{n}{n+1}$ puis la compiler. Définir ensuite la fonction g par $g(n) = \frac{n+1}{n+2}$ puis la compiler. Vous obtiendrez ceci :



Cliquez sur « J'ai terminé » pour revenir dans le tableur et construire une colonne « n » contenant les entiers de 0 à 15 puis à sa droite une colonne à partir de la formule =f(A1) et une autre colonne à partir de la formule =g(A1).

On obtient ensuite les valeurs arrondies à 10^{-6} avec l'instruction =evaluer(B1;6) (à adapter bien sûr !)

2. Peut-on répondre au problème posé ?
3. Calculer dans une autre colonne la valeur exacte de $\frac{n}{n+1} - \frac{n+1}{n+2}$. Commenter les résultats.

aide : prendre exemple sur le calcul exact de $\frac{n}{n+1}$ et $\frac{n+1}{n+2}$.

4. Sous la cellule contenant $n = 15$, ajouter une cellule contenant "=" et recopier les formules calculant $\frac{n}{n+1}$, $\frac{n+1}{n+2}$ et

$$\frac{n}{n+1} - \frac{n+1}{n+2}$$

Répondre alors rigoureusement au problème posé.

2. Des activités pour les élèves

a) Somme de trois entiers consécutifs

Cette activité va mettre en évidence tout l'intérêt de la factorisation. On laisse la machine effectuer des calculs algébriques qui dépassent parfois le niveau collège pour se concentrer sur le raisonnement. On aboutira à une généralisation de cette propriété dans certains cas.

On veut montrer que la somme de trois entiers consécutifs est toujours divisible par 3.

1. Créer une feuille de calcul sur le modèle ci-dessous et utiliser uniquement la fonction "csomme" pour vérifier cette propriété sur quelques exemples (il est possible, avec quelques formules bien choisies de remplir tout le tableau en donnant uniquement un nombre de départ dans la cellule A2) :

	A	B	C	D
1	Premier entier	Deuxième entier	Troisième entier	Somme des trois entiers
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				

2. A la dernière ligne du tableau, remplacer le « premier entier » par n .
- Qu'obtient-on comme « somme des trois entiers » ?
 - A l'aide de la fonction "cquotient", diviser cette somme par 3 puis simplifier le résultat (icône "simplifier").
 - Pourquoi ce résultat est-il entier ?
 - Comment l'obtenir par le calcul ?
 - A-t-on ainsi démontré cette propriété ?

Production attendue :

	A	B	C	D	E
1	Premier entier	Deuxième entier	Troisième entier	Somme des trois entiers	
2	1	2	3	6	
3	2	3	4	9	
4	3	4	5	12	
5	4	5	6	15	
6	5	6	7	18	
7	6	7	8	21	
8	7	8	9	24	
9	8	9	10	27	
10	9	10	11	30	
11	10	11	12	33	
12	11	12	13	36	
13	12	13	14	39	
14	13	14	15	42	
15	14	15	16	45	
16	15	16	17	48	
17	16	17	18	51	
18	17	18	19	54	
19	18	19	20	57	
20	19	20	21	60	
21	20	21	22	63	
22	21	22	23	66	
23	22	23	24	69	
24	23	24	25	72	
25	n	n+1	n+2	3*n+3	n+1
26					
27					

On veut savoir si, plus généralement, la somme de 2 entiers consécutifs est divisible par 2, la somme de 4 entiers consécutifs est divisible par 4, etc...

1. Cette propriété est-elle vraie pour la somme de 2 entiers consécutifs ? pour 4 ?
2. A l'aide de l'éditeur de fonctions (icône "CAS") compléter puis compiler l'algorithme suivant qui calcule la somme de p entiers consécutifs en commençant à n :

```

fonction pSommes(n,p)
  local s:=0;
  pour k de n jusque . . . . . faire
    . . . . .
  fpour
  retourne s
ffonction
    
```

A l'aide de la ligne de commande, utiliser la fonction pSommes pour calculer quelques sommes de 2 puis 4 entiers consécutifs pour confirmer la réponse à la question 1.

3. Utiliser encore cette fonction pour calculer quelques sommes de 5 entiers consécutifs. Que constate-t-on ?
4. Reproduire puis remplir le tableau suivant en utilisant la fonction pSommes :

	A	B
1	Somme de p entiers consécutifs.	
2	p=5	
3		
4	Premier entier	Somme
5	1	
6	2	
7	3	
8	4	
9	5	
10	6	
11	7	
12	8	
13	9	
14	10	
15	11	

Cela suffit-il à démontrer la propriété pour 5 entiers consécutifs ?

5. Remplacer un des entiers par n . Dans la colonne C, demander la simplification du résultat obtenu. Comment conclure rigoureusement quant à la somme de 5 entiers consécutifs ?
6. En testant des valeurs de p impaires, vérifier que la propriété reste vraie.
7. En testant des valeurs de p paires, vérifier que la propriété est toujours fausse.

Production attendue :

	A	B	C
1	Somme de p entiers consécutifs.		
2	p=5		
3			
4	Premier entier	Somme	
5	1	15	
6	2	20	
7	3	25	
8	4	30	
9	5	35	
10	6	40	
11	7	45	
12	8	50	
13	9	55	
14	10	60	
15	11	65	
16	12	70	
17	13	75	
18	14	80	
19	15	85	
20	16	90	
21	17	95	
22	18	100	
23	19	105	
24	n	$\frac{((n+4+1)^2-n-4-1)}{2}-\frac{(n^2-n)}{2}$	Simplification : 5*n+10
25			Factorisation : 5*(n+2)
26			

Prolongement : démonstration des conjectures

8. A l'aide de l'éditeur de fonctions (icône "CAS") compiler cette nouvelle définition de la fonction pSommes :

$$pSommes(n, p) := \{ return (somme(k, k, n, n+p-1)) \}$$
 et vérifier sur des exemples qu'elle calcule toujours la somme de p entiers consécutifs en commençant à n . Les curieux pourront regarder la syntaxe de cette fonction "somme" en tapant ?somme dans la ligne de commande.
9. Entrer la valeur $2q + 1$ dans la cellule B2.
 Calculer la forme factorisée du résultat simplifié (utiliser la fonction "factoriser").
 Peut-on conclure ?
10. Terminer l'étude de cette propriété pour les nombres p pairs.

Production attendue :

	A	B	C
1	Somme de p entiers consécutifs.		
2		p= 2q+1	
3			
4	Premier entier	Somme	
5	1	$((1+2*q+1)^2-1-2*q-1)/2$	
6	2	$((2+2*q+1)^2-2-2*q-1)/2-1$	
7	3	$((3+2*q+1)^2-3-2*q-1)/2-3$	
8	4	$((4+2*q+1)^2-4-2*q-1)/2-6$	
9	5	$((5+2*q+1)^2-5-2*q-1)/2-10$	
10	6	$((6+2*q+1)^2-6-2*q-1)/2-15$	
11	7	$((7+2*q+1)^2-7-2*q-1)/2-21$	
12	8	$((8+2*q+1)^2-8-2*q-1)/2-28$	
13	9	$((9+2*q+1)^2-9-2*q-1)/2-36$	
14	10	$((10+2*q+1)^2-10-2*q-1)/2-45$	
15	11	$((11+2*q+1)^2-11-2*q-1)/2-55$	
16	12	$((12+2*q+1)^2-12-2*q-1)/2-66$	
17	13	$((13+2*q+1)^2-13-2*q-1)/2-78$	
18	14	$((14+2*q+1)^2-14-2*q-1)/2-91$	
19	15	$((15+2*q+1)^2-15-2*q-1)/2-105$	
20	16	$((16+2*q+1)^2-16-2*q-1)/2-120$	
21	17	$((17+2*q+1)^2-17-2*q-1)/2-136$	
22	18	$((18+2*q+1)^2-18-2*q-1)/2-153$	
23	19	$((19+2*q+1)^2-19-2*q-1)/2-171$	Simplification :
24	n	$((n+2*q+1)^2-n-2*q-1)/2-(n^2-n)/2$	$2*n*q+n+2*q^2+q$
25			Factorisation :
26			$(2*q+1)*(n+q)$
27			

b) un tour de magie

Julien aime faire ce tour de magie à ses camarades :

Il leur demande :

- a) de choisir un nombre (noté N) compris entre 1 et 987 dont tous les chiffres sont différents.
- b) d'écrire tous les nombres possibles en mélangeant les chiffres de N.
- c) d'additionner tous ces nombres.
- d) de diviser le résultat obtenu par la somme des chiffres de N

Après avoir demandé combien N a de chiffres, il est capable de dire le résultat obtenu à la dernière étape.

Pour comprendre comment ce tour fonctionne, nous allons travailler d'abord avec un nombre N compris entre 10 et 98 dont tous les chiffres sont différents (en choisir un).

1. Expérimentation :
 - a) Ouvrir une nouvelle feuille de calcul. Dans une cellule inscrire le chiffre des dizaines de N et dans une autre le chiffre des unités. Écrire en colonne les nombres répondant à la demande b de Julien (attention, les nombres de cette colonne doivent varier lorsque N change).
 - b) Calculer dans une cellule la somme de tous ces nombres (demande c)
 - c) Afficher dans une cellule le résultat final (demande d)
 - d) En faisant varier le chiffre des unités et des dizaines de N, conjecturer comment Julien peut deviner le résultat pour n'importe quel nombre N à deux chiffres.
2. Validation de la conjecture :

Nous allons transformer la feuille de calcul pour travailler sur un nombre N quelconque, écrit sous la forme " du " où d désigne le chiffre des dizaines de N et u le chiffre des unités.

 - a) entrer la formule " d " dans la cellule du chiffre des dizaines de N et " u " dans celle des unités.
 - b) les fonctions +, *, / du tableur ne savent pas travailler avec des expressions littérale. Modifier la feuille de calcul pour

utiliser les fonctions : `csomme`, `cproduit` et `cquotient` (par exemple, pour additionner deux expressions, il faut entrer : `=csomme(expression1;expression2)`)

c) expliquer pourquoi le dernier résultat trouvé valide la conjecture émise en 1d.

3. Reprendre ce travail pour un nombre N à 3 chiffres.
4. Rédiger un algorithme qui détaille ce que doit répondre Julien pour deviner le résultat lorsque N est compris entre 1 et 987.

Prolongements :

5. Julien peut-il demander un nombre entre 1 et 9876 ? Si oui, que devient l'algorithme de la question 4 ?
6. Pourquoi faut-il que tous les chiffres soient différents ?

c) A la recherche d'un minimum

Dans un rectangle ABCD, on place un point I sur le segment [AB]. On trace ensuite un quadrilatère IJKL tel que $AI=BJ=CK=DL$ et tel que $J \in [BC]$, $K \in [CD]$ et $L \in [DA]$

1. Construire cette figure dans un logiciel de géométrie dynamique
2. Conjecturer la nature du quadrilatère IJKL
3. Afficher l'aire A de la surface du quadrilatère IJKL et conjecturer les variations de A en fonction de la longueur AI.

4. On va étudier avec le tableur les variations de A .
On appelle L la longueur du rectangle et l sa largeur.

On pose $AI=x$ cm.

a) Déterminer une expression de l'aire A en fonction de x , L et l .

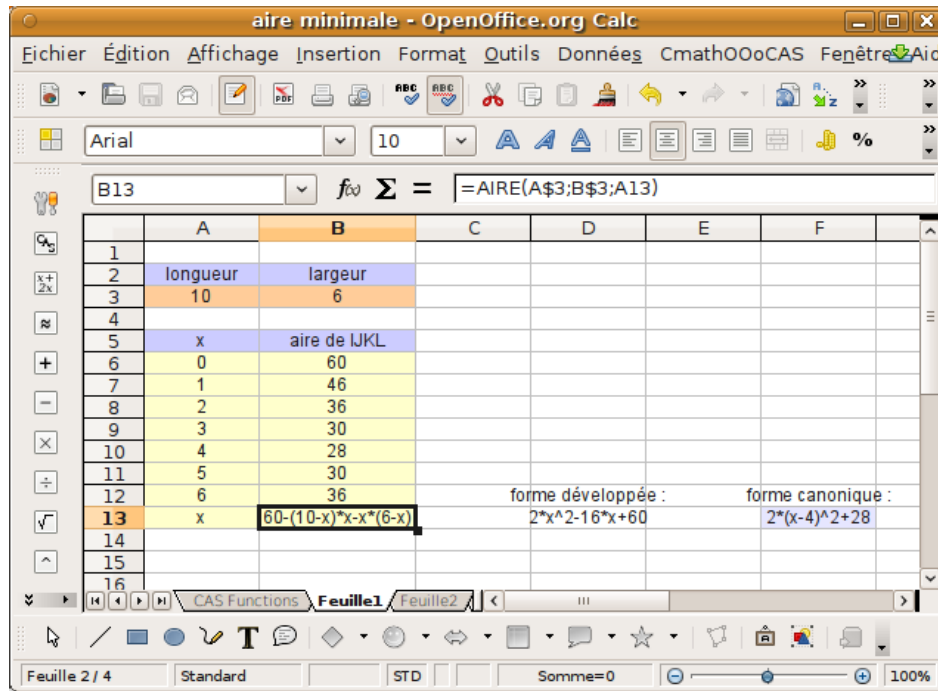
b) Avec l'éditeur de fonctions, définir une fonction « aire » de la façon suivante :

```
aire(L, l, x) := {
  return( mettre ici l'expression trouvée à la question a );
}
```

c) pour un rectangle de longueur 10 cm et de largeur 6 cm, construire une feuille de calcul calculant A , pour des valeurs de x comprise entre 0 et 6 cm (inscrire les valeurs 10 et 6 dans des cellules et y faire référence lors de l'appel à la fonction `aire`)

5. Pour quelle valeur (notée x_{min}) de x obtient-on un minimum ? Combien vaut-il ? Est-on certain d'avoir trouvé le minimum ?
6. Ajouter une ligne affichant A pour x cm. Quel type d'expression obtient-on pour A ?
7. Demander l'affichage de la forme canonique de A et conclure quant à la valeur de x_{min} .
8. En faisant varier la longueur et la largeur du rectangle ABCD, conjecturer une relation entre x_{min} , L et l .
9. En étudiant la forme canonique affichée, démontrer la conjecture émise à la question précédente en donnant au tableur les valeurs L et l comme longueur et largeur.

production attendue à la question 7 :



3. Des problèmes dont l'activité est à créer

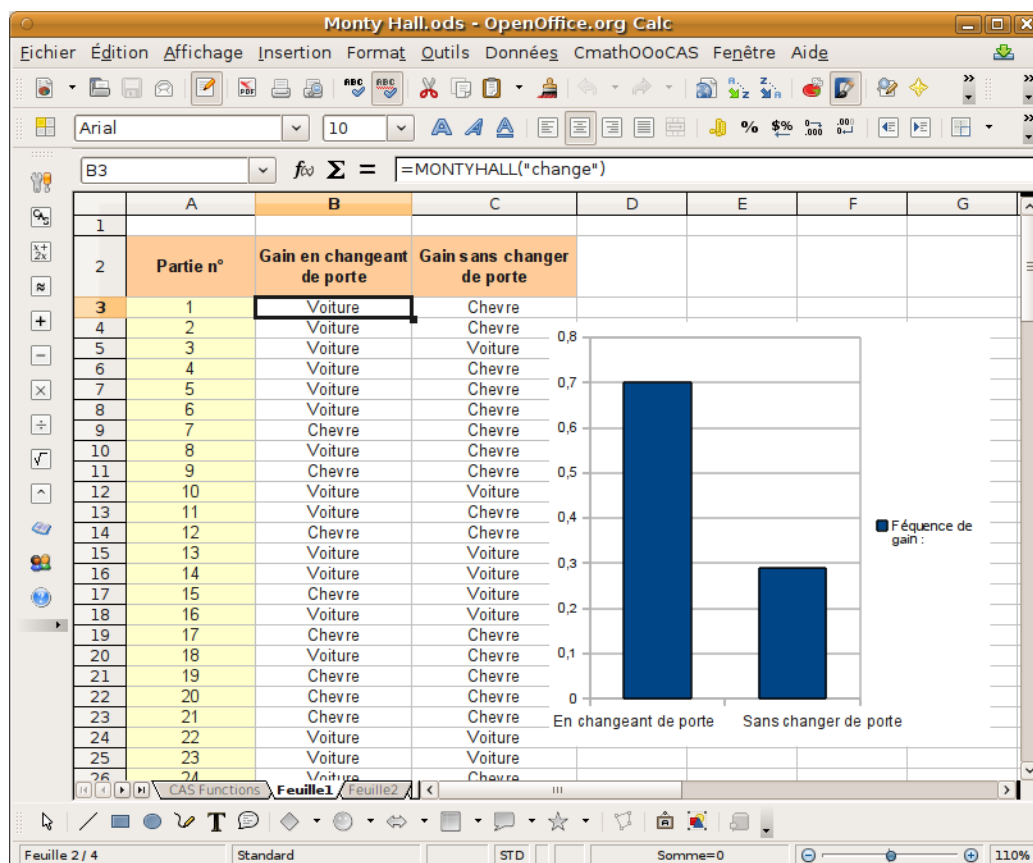
a) le test de Monty Hall

Le nom de ce problème mathématique vient du nom de l'animateur d'origine canadienne Monty Hall qui a présenté ce jeu aux États-Unis pendant treize ans. Voici son déroulement :

Le jeu oppose un présentateur à un candidat (le joueur). Ce joueur est placé devant trois portes fermées. Derrière l'une d'elles se trouve une voiture et derrière chacune des deux autres se trouve une chèvre. Il doit tout d'abord désigner une porte. Puis le présentateur ouvre une porte qui n'est ni celle choisie par le candidat, ni celle cachant la voiture (le présentateur sait quelle est la bonne porte dès le début). Le candidat a alors le droit ou bien d'ouvrir la porte qu'il a choisie initialement, ou bien d'ouvrir la troisième porte.

La question que l'on se pose est : Quelle est la stratégie qui offre le plus de chance de gagner la voiture ?

Remarques : si on veut travailler avec une fonction simulant une partie de ce jeu, l'algorithme n'est pas si difficile à écrire, mais sa traduction l'est davantage. Il est possible d'utiliser la fonction `MontyHall(change)`, disponible dans les « modèles de fonctions » de l'éditeur de fonctions. Elle retourne le gain du joueur : « voiture » ou « chèvre » en tenant compte de l'argument « change » en entrée qui permet d'indiquer à la fonction si le joueur change d'avis (`change="change"`) ou pas (`change` vaut n'importe quoi d'autre). La production attendue pourrait ressembler à cela (100 parties) :



b) un autre tour de magie

Écrire une activité permettant d'expliquer le tour de magie suivant :

« Demandez à une personne d'écrire sur une feuille de papier les réponses à ces questions :

1. Combien de fois par semaine aimes-tu manger du chocolat ? (le chiffre doit être compris entre 1 et 9).
2. Multiplie ce chiffre par 2.
3. Ajoute 5.
4. Multiplie le résultat par 50.
5. Si tu as déjà fêté ton anniversaire en 2010 ajoute 1760. Si tu n'as pas encore fêté ton anniversaire cette année ajoute 1759.
6. Maintenant retranche l'année de ta naissance (nombre de quatre chiffres).

Le résultat est un nombre de trois chiffres. Demandez ce nombre à la personne. Vous pouvez alors lui annoncer :

- Le nombre de fois qu'elle aime manger du chocolat par semaine (c'est le premier chiffre).
- Son âge (c'est le nombre formé par les deux derniers chiffres) ».

Remarque : plusieurs approches sont possibles. construction d'une fonction formelle après avoir élaboré un algorithme, construction d'une feuille de calculs faisant apparaître les étapes successives...

c) composées de fonctions

La décomposition d'une fonction f en fonction élémentaires permet de justifier de nombreuses propriétés de f (ensemble de définition, variations, limites...). Pourtant, la notion de composée d'une fonction, opération souvent confondue avec la multiplication au moment de l'apprentissage, nécessite une mise en pratique importante pour qu'elle soit bien assimilée. Il est facile avec la fonction « substituer » de CmathOOoCAS de construire une activité-jeu demandant de trouver la décomposition de fonctions données à l'avance en utilisant uniquement les fonctions de référence.

Une fois la méthode bien comprise, on peut ajouter des questions concernant la commutativité, la recherche d'ensemble de définition, etc...

d) les fractions égyptiennes ; algorithme de Fibonacci-Sylvester

D'après Wikipédia : « Une **fraction égyptienne** est une somme de fractions unitaires, c'est-à-dire de fractions qui ont des numérateurs égaux à 1 et des dénominateurs entiers positifs, avec ces dénominateurs tous différents les uns des autres. Il peut être montré que tous les nombres rationnels positifs peuvent être écrits sous cette forme et ce, d'une *infinité* de façons différentes. Par exemple : $\frac{2}{5} = \frac{1}{5} + \frac{1}{6} + \frac{1}{30}$. (...) La notation sous forme de fractions égyptiennes a de plus été utilisée pendant la période grecque et même au Moyen Âge ».

En 1201, Fibonacci (Léonard de Pise 1175-1250) trouve un algorithme, qu'il décrit sans démonstration dans son livre « Liber abaci » paru en 1202. :

« Soustraire à la fraction donnée la plus grande fraction égyptienne possible, répéter l'opération avec la nouvelle fraction, et ainsi de suite jusqu'à ce que l'opération donne une fraction égyptienne. »

Cet algorithme donne une décomposition en fractions égyptiennes tout rationnel $r \in [0 ; 1]$ Cette méthode a été redécouverte en 1880 par James Sylvester. On appelle cette méthode « algorithme glouton » de Fibonacci-Sylvester. James Sylvester a démontré que l'algorithme donne bien une décomposition en un nombre fini de fractions unitaires.

Construire une activité mettant en œuvre cet algorithme pour décomposer un nombre rationnel en fractions égyptiennes. On pourra mettre en évidence que cette décomposition n'est pas unique ni toujours la plus courte. Les dénominateurs sont parfois très grands, dépassant les capacités d'un tableur « classique ». Il n'y aura bien sûr pas de problème avec OpenOffice+CmathOOoCAS. Par exemple, cet algorithme donne :

$$\frac{5}{121} = \frac{1}{25} + \frac{1}{757} + \frac{1}{763309} + \frac{1}{873960180913} + \frac{1}{1527612795642093418846225} \text{ alors que } \frac{5}{121} \text{ s'écrit aussi } \frac{1}{33} + \frac{1}{121} + \frac{1}{363}.$$

	A	B	inverses :
1	5/121		
2	fractions égyptiennes :	restes :	
3	1/25	5/121	
4	1/757	4/3025	
5	1/763309	3/2289925	
6	1/873960180913	2/1747920361825	873960180913
7	1/1527612795642093418846225	1/1527612795642093418846225	1527612795642093418846225
8		0	infinity
9			
10	Décomposition :		
11	5/121 =	1/25	+ 1/757

Illustration 5: Merci à Alain Busser qui m'a donné cette idée et construit le document OOoCalc+CmathOOoCAS correspondant

e) nombres équirépartis, nombres univers

Définition : On dit qu'un nombre réel x est **équiréparti** si, dans la suite infinie constituée des décimales de x (parfois il peut y avoir beaucoup de zéros !), chacun des chiffres de 0 à 9 a une fréquence d'apparition de $\frac{1}{10}$.

La fonction `evaluer(nombre;nb_decimales)` permet d'obtenir autant de décimales que l'on souhaite d'un nombre donné. Exploiter cette fonction pour étudier si certains nombres sont équirépartis ou non (nombres entiers, décimaux, nombres de la forme $\frac{1}{n}$, π , $\sqrt{2}$, etc...)

On peut également jouer avec les **nombre « univers »** : tout nombre réel dont la suite des décimales contient toute suite finie choisie à l'avance. (Le nom donné à ces nombres se justifie par le fait que, à condition de numériser l'information, le développement décimal d'un tel nombre renferme aussi bien le prochain tirage du loto, le prochain sujet du contrôle de mathématiques, l'édition complète des Misérables (1800 pages dans la collection La Pléiade), l'œuvre complète de Mozart (ou des Beatles), la toile de La Joconde pixel par pixel, mais aussi tout le code génétique de chacun d'entre nous ... ! Un tel nombre renferme donc dans son écriture décimale tout ce qui existe, existera ou a jamais existé (en nombre fini) dans l'univers.)