

Le développement informatique réalisé pour la mise en place de la base raisonnée d'exercices de mathématiques pour le DEUG

François Dagorn
28 janvier 2005

1 Introduction

La base raisonnée d'exercices de mathématiques pour le DEUG a pour but de permettre à des étudiants de réaliser des exercices en recevant des aides contextuelles. Les étudiants sont invités à choisir des exercices par niveau de difficulté, thème, ... ils peuvent même préciser des difficultés particulières qu'ils souhaitent éviter.

D'un point de vue technique, on peut noter deux contraintes importantes : l'accès au service doit s'effectuer à partir d'un navigateur WWW, les exercices sont au format \LaTeX et doivent être visionnés avec une typographie adaptée aux mathématiques.

À l'issue de diverses réunions avec les enseignants responsables du projet de constitution d'une base raisonnée d'exercices de mathématiques pour le DEUG, les décisions suivantes ont été prises :

- la réalisation du projet est découpée en deux phases : d'une part la mise à jour de la base (l'interface enseignant) et d'autre part l'utilisation de la base (l'interface étudiant) ;
- la plupart des exercices *existent déjà* et ont été rédigés dans le format du traitement de texte scientifique \LaTeX . De plus les enseignants souhaitent vivement conserver ce mode opératoire. Pour des besoins de présentation et de réutilisation, les exercices seront découpés selon un schéma élaboré par les enseignants. Un exercice est composé *d'atomes* (ils sont en général rédigés au format \LaTeX), qui ont la forme suivante :
 - un chapitre ;
 - un titre ;
 - un numéro ;
 - un énoncé ;
 - un ou plusieurs thèmes ;
 - un niveau de difficulté (facile, ..., très difficile) ;
 - une ou plusieurs natures de la tâche à réaliser (que demande t-on à l'étudiant ?) ;
 - une ou plusieurs difficultés particulières (que l'étudiant pourra demander d'éviter) ;
 - un ou plusieurs éléments de cours utilisables (des rappels de connaissances pouvant aider pour réaliser le travail demandé) ;
 - une ou plusieurs méthodes et techniques pour trouver la solution ;
 - d'éventuelles indications particulières ;
 - d'éventuels éléments supplémentaires de solutions et résultats ;
 - d'éventuelles idées à retenir ;

- une éventuelle aide graphique (l'URL d'une image ou d'une animation).
- Les exercices seront traduits en HTML à l'aide d'un traducteur. Diverses solutions ont été testées, $\text{T}\text{E}\text{X}_4\text{ht}$ a été choisi car il procure un rendu typographique assez proche de la qualité recherchée. $\text{T}\text{E}\text{X}_4\text{ht}$ peut être configuré pour produire du HTML 3.2 ou du HTML 4.0 avec feuille de styles associée, c'est cette dernière option qui a été retenue car elle produit un code HTML compact et bien digéré par les navigateurs les plus couramment utilisés à l'université. $\text{T}\text{E}\text{X}_4\text{ht}$ permet en outre de traduire les exercices au format MathML. Cette possibilité intéressante n'a pas été pour l'instant explorée (les enseignants et les salles libre service des étudiants sont équipés de navigateurs *netscape 4.x* qui n'autorisent pas l'usage du format MathML). Conserver les exercices au format $\text{L}\text{A}\text{T}\text{E}\text{X}$ a été également envisagé, un *plug-in* est alors nécessaire pour les visionner, la non-disponibilité de cet additif sur l'ensemble des postes ciblés a rendu ce choix peu attractif. En prolongement du projet, il serait intéressant de traduire les exercices en HTML 4.0 ainsi qu'en MathML, ils pourraient être ainsi automatiquement visionnés par les navigateurs *modernes* avec un rendu typographique parfait.

2 Le format de la base raisonnée d'exercices de mathématiques

Le développement du contenu de plusieurs chapitres est prévu, une cinquantaine d'exercices seront proposés pour chacun d'entre eux. La base contiendra donc à terme environ 400 ou 500 exercices. Comme indiqué précédemment, la structure de la base raisonnée est assez simple, un exercice est composé de quelques champs, certains pouvant être présents plusieurs fois (plusieurs thèmes pour un exercice par exemple). Compte tenu du faible volume d'exercices à gérer et de la fréquence très probablement légère des opérations de mise à jour (les exercices sont stables), il n'est pas nécessaire d'utiliser un SGBD comme support de la base raisonnée, elle est donc supportée par un seul fichier au format XML.

L'adoption du standard XML (adapté au traitement de l'information structurée) ouvre la porte vers un grand nombre de fonctionnalités; la figure 1 a, par exemple, été réalisée en ouvrant le document XML supportant la base avec le navigateur Internet Explorer.

3 Les opérations liées à la mise à jour d'un exercice

1. La création ou la modification d'un exercice dans la base est réalisée au moyen d'une interface WWW (cf. figure 2). Les enseignants renseignent les champs spécifiés en $\text{L}\text{A}\text{T}\text{E}\text{X}$ en effectuant des copier/coller depuis leurs originaux et positionnent des options de menus déroulants pour les caractéristiques communes (thèmes, nature de la tâche, difficultés particulières, ...).
2. La validation de la saisie d'un exercice entraîne les opérations suivantes :
 - mise à jour de la structure XML contenant les exercices ;
 - traduction en HTML :
 - création, si nécessaire, d'un répertoire pour recevoir la traduction ;

- création d’un fichier au format L^AT_EX (*exo.tex*) qui va servir de source au processus de traduction. On reconstitue le texte d’un exercice en recollant ses *atomes*. Des commentaires sont ajoutés dans le document HTML produit, ils seront par la suite utilisés pour accéder à la traduction HTML des *atomes* (titre, énoncé, thèmes, ...). T_EX₄ht est lancé dans un processus séparé car l’opération peut être très longue (le temps de traduction dépend directement du nombre d’équations présentes dans le document) et de plus, après de quelques secondes, on peut déterminer si la traduction sera correcte ou non. Dès que possible, on passe à l’émission du résultat tandis que la traduction des équations en images va s’opérer en arrière plan. Lorsque la phase de traduction est terminée, divers traitements sont effectués sur le code HTML produit (pour le rendre plus lisible ou corriger certaines anomalies fonctionnelles, comme celle de l’extension *hyperref* pour L^AT_EX qui ne permet pas de construire des liens hypertextes ouvrant la cible dans une fenêtre séparée).
 - émission du résultat sous la forme de la liste des exercices présents pour le chapitre considéré (cf. figure 3) en précisant, pour chacun d’entre eux, le résultat du processus de traduction. La présence d’une icône jaune seule devant un exercice indique que la traduction s’est correctement déroulée. En cliquant sur l’icône, l’enseignant accède au rendu HTML de l’exercice (cf. figure 4). La présence d’une icône rouge seule signale que la traduction n’a pu s’opérer (erreur dans le source L^AT_EX), en cliquant sur l’icône rouge, l’enseignant visualise les erreurs produites par la traduction. La présence de deux icônes (une rouge et une jaune) indique que l’exercice est correctement codé en L^AT_EX mais que la traduction a échoué (insuffisances ou bogues du traducteur). Dans ce cas, il faut trouver une autre formulation L^AT_EX pour la séquence concernée.
3. L’ensemble des exercices est contenu dans un seul document XML qui doit être protégé contre les accès concurrents en mise à jour. Lorsqu’un enseignant souhaite modifier ou supprimer un exercice, il fait appel à l’interface WWW appropriée (cf. figure 2) pour indiquer ses modifications. L’affichage de l’interface WWW est accompagnée par un ticket qui précise l’état du document XML contenant les exercices (la base). Lorsque l’enseignant a terminé ses modifications, il demande à valider les changements, le ticket courant d’état de la base est alors comparé au ticket initialement reçu. Si les deux tickets sont identiques, la mise à jour est acceptée, sinon l’enseignant est invité à soumettre à nouveau les modifications. Cette opération évite qu’un enseignant détruise un exercice qu’un collègue souhaite modifier ou que deux enseignants modifient le même exercice (les modifications de l’un pouvant être écrasées par celles de l’autre). Ce contrôle ne concerne pas les opérations de création d’exercices pour lesquelles il n’y a pas d’ambiguïtés.

De plus, l’accès au contenu de la base est organisé en section critique par la création et la libération d’un verrou. Lorsque un verrou est détecté, le processus suivant est répété au plus dix fois :

- attente d’une seconde ;
- le verrou est-il toujours présent ?
 - si oui on attend à nouveau ;
 - si non un verrou est posé et le traitement peut commencer.

À l’issue de dix tentatives infructueuses, on effectue tout de même le traitement, le verrou trouvé est très certainement le résultat d’une opération mal terminée (plantage du programme ou du système à l’intérieur de la section critique).

Ce procédé ultra simple est adapté aux volumes à traiter. Si 200 enseignants devaient mettre à jour une base de 100 000 exercices, il faudrait évidemment envisager une solution plus élaborée.

4 La consultation de la base d'exercices

Un étudiant peut consulter la base en effectuant une recherche préalable par mots-clés ou en demandant à visionner les exercices auxquels il a accédé lors d'une consultation antérieure.

Dans les deux cas, la présentation des exercices est identique, seuls les deux modes de sélection d'exercices diffèrent. Dans le cas du choix par mots-clés, on passe par une étape d'affichage des critères de sélection (cf. figure 5) alors que si l'étudiant souhaite accéder aux exercices déjà vus, c'est la liste de ceux-ci qui sert de sélection.

Lorsqu'un étudiant indique un nom d'utilisateur, il a la possibilité d'accéder rapidement aux exercices qu'il a déjà traités, de plus les enseignants peuvent suivre son activité (cf. figure 9). Une structure de données permanente est gérée à cet effet, elle indique quels sont les exercices consultés par un étudiant.

Lors de l'affichage partiel (résultat d'une sélection, cf. figure 6) ou complet (demande d'utilisation, cf. figure 7) d'un exercice, il convient d'accéder à la traduction HTML des *atomes* qui le composent. Comme indiqué en 3.2, on dispose, pour chaque exercice, d'un fichier HTML (*exo.html*) qui est le résultat de sa traduction par $\text{T}_{\text{E}}\text{X}_{4}\text{ht}$. Des commentaires insérés à l'aide de $\text{T}_{\text{E}}\text{X}_{4}\text{ht}$ permettent d'accéder à la traduction des *atomes* comme indiqué dans l'exemple suivant :

```
<!-- <enonce> -->
<table width='90%' align='center'>
<tr><td> Etudier la convergence de la suite&nbsp;&nbsp;&nbsp;;:

<!--l. 63--><p class="indent">
</td></tr></table>
<!-- </enonce> -->

  <h2> <span class="textcolor3">Th&egrave;me</span>
<!--l. 69--><p class="indent">  </h2>
<!-- <theme> -->
<ul class="itemize1">
<li class="itemize"><a
href="/tdmath/donnees/ParamHTML/Les%20suites/.../cst.html"
onclick="opw()" target="param" >Les premiers exercices</a></li>
</ul>
<!--l. 73--><p class="nopar">
<!--l. 75--><p class="indent">
<!-- </theme> -->
.....
<!-- <titre> -->

<!--l. 83--><p class="indent">
<!-- </titre> -->
```

5 Le paramétrage d'un chapitre

Un certain nombre de champs qui constituent un exercice sont directement dépendants du chapitre concerné. Les thèmes *Suites de Cauchy* ou *Suites monotones bornées* sont dédiés au chapitre *les suites*, ils n'apparaissent pas dans le chapitre *Algèbre linéaire*. Il en est de même pour les *difficultés particulières* ou les *nature de la tâche*, de plus ces paramètres donnent lieu à une définition qui doit pouvoir être consultée par un étudiant souhaitant être renseigné. La définition d'un paramètre (thème, difficulté, ...) est donc la suivante :

- le titre qui est présenté dans les interfaces *enseignants* de sélection (créer ou modifier un exercice). Un titre peut, s'il comporte des symboles mathématiques, être défini deux fois : en version textuelle pour apparaître dans les menus de mise à jour et en version $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ pour être convenablement présenté aux étudiants ;
- un texte qui est au format $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ et qu'il convient de traduire en HTML en utilisant $\text{T}_{\text{E}}\text{X}_4\text{ht}$. C'est ce texte qui est présenté lorsqu'un étudiant clique sur un lien évoquant un thème, une difficulté, ...

L'ensemble des paramètres utiles à la constitution d'un chapitre ont été regroupés dans un document XML (cf. figure 8) qui est joint à celui supportant la base d'exercices lors des différents traitements effectués par des programmes XSLT.

Il n'a pas été développé de programme pour effectuer la mise à jour du document XML supportant les paramètres, **il faudra le faire**. Pour l'instant les modifications sont directement effectuées à l'aide d'un éditeur de texte. Ce procédé est évidemment incompatible avec les principes d'XML qui impose qu'un document ne contienne pas les caractères $<$ et $\&$ pourtant très fréquents dans un texte source au format $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ (ils doivent être *échappés* par $\&\text{amp}$; et $\&\text{l t}$;). Avant tout usage par un programme XSLT, les éventuels caractères parasites sont donc traités. Il n'en reste pas moins qu'en dehors du contexte de la base d'exercices, le document XML supportant les paramètres de chapitre, **n'est pas bien formé**, et ne peut donc être utilisé en dehors de ce contexte (la figure 8 n'aurait pu s'effectuer sans *retouches* !).

Compte tenu de l'absence de programme de mise à jour des paramètres de chapitre, le processus de traduction en HTML du texte des thèmes, ... est automatiquement démarré (si nécessaire) lorsqu'un enseignant appelle l'interface de création ou de mise à jour d'exercices. La traduction s'opère globalement (pour tous les thèmes, ...) et dans un processus opérant automatiquement en arrière plan. Si des erreurs sont détectées, elles seront signalées par une icône rouge qui aiguillera vers le ou les paramètres erronés.

6 Le suivi de l'activité des étudiants

Les étudiants volontaires peuvent donner un nom lorsqu'ils consultent la base d'exercices. Ils ont alors accès à la liste des exercices qu'ils ont déjà consultés lors de sessions antérieures. En outre, les enseignants peuvent alors suivre leurs activités dans le but d'améliorer la pertinence de la base. L'interface de mise à jour de la base d'exercices permet d'accéder à cette fonctionnalité. Après avoir sélectionné le nom d'un étudiant, l'enseignant demandeur obtient le résultat présenté par la figure 9.

Le résultat est obtenu en cumulant la trace des consultations dans un fichier géré par *syslog*. L'exploitation est ensuite triviale, il reste simplement à déterminer la fréquence de remise à zéro du fichier cumulant les traces d'accès.

7 Conclusion

Dans son état actuel la base d'exercices est consultable par <http://tdmath.univ-rennes1.fr> elle a été développée en 3,5 mois (de septembre à décembre 2002).

Comme indiqué en section 5 il est nécessaire de continuer le développement pour réaliser un programme de mise à jour des paramètres utiles à la constitution d'un chapitre. Ce développement rendra les enseignants totalement autonomes, actuellement ils doivent passer par l'intermédiaire du développeur qui gère les mises à jour de la structure XML contenant les définitions de thèmes, ... associés aux chapitres.

Il convient également d'assurer la traduction des exercices de $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ en MathML pour assurer une qualité typographique parfaite. La conversion en MathML effectuée par $\text{T}_{\text{E}}\text{X}_{4}\text{h}$ est très rapide (il n'est pas nécessaire de convertir les expressions mathématiques en images), un exemple de conversion d'un gros document $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ est disponible à l'adresse <http://tdmath.univ-rennes1.fr/test/exo.xml>¹.

Lorsque cette modification sera opérationnelle, le navigateur client sera automatiquement détecté, il recevra un code approprié (HTML ou MathML).

Diverses interventions dans le code sont aussi nécessaires pour obtenir une plus grande indépendance entre les traitements et la présentation des résultats.

¹ Pour le visionner il convient de disposer du navigateur Mozilla 1.0 ou ultérieur

A La structure de l'implémentation

L'application est architecturée autour de deux programmes écrits en Perl : *braise.cgi* (interface étudiant) et *mbraise.cgi* (interface enseignant). Ils s'appuient sur les modules Perl *XML : :LibXSLT* et *XML : :LibXML* qui interfacent la bibliothèque Libxml2² du projet Gnome.

- **répertoire d'installation**
- index.html
- braisefl.cgi (Perl)
- braise.cgi (Perl)
- ecran4.xsl (XSLT)
- **prot**
 - .htaccess
 - mbraise.cgi (Perl)
 - braise.pm (Perl)
 - log.pm (Perl)
 - traque.pm (Perl)
 - conf.cfg (configuration de T_EX₄ht)
 - liste.xsl (XSLT)
 - affich.xsl (XSLT)
- **donnees**
 - Exos.xml, Exos.xml.0, Exos.xml.1, Exos.xml.2, ...
 - Param.xml
 - DejaVus.txt
- **HTML**
 - **12-11-2002-14 :23 :53**
 - exo.tex
 - exo.html
 - exo.css
 - erreur.txt
 - exo0.gif, exo1.gif, exo2.gif, ...
 - **16-12-2002-16 :35 :02**
 - ...
- **ParamHTML**
 - Date
 - **Algèbre**
 - **Les suites**
 - **con**
 - **dif**
 - **nat**
 - **met**
 - **theme**
 - **Calcul de la valeur approchée**
 - cst.tex

² <http://xmlsoft.org>

- cst.html
- cst.css
- erreur.txt
- cst0.gif, cst1.gif, cst2.gif, ...
- **Cesaro et compagnie**
- ...

- **Verrou**

Le répertoire d'installation de l'application contient le point d'entrée de la partie consultation (*index.html*) ainsi que les programmes *braise.cgi* (interface étudiant) et *braisefl.cgi* (chargement des différentes frames créées par *index.html*). Le programme XSLT *ecran4.xsl* est appelé par *braise.cgi*, il affiche les critères de sélection d'exercices par mots-clés (cf. figure 5).

Le répertoire **prot** contient le nécessaire à la mise à jour de la base d'exercices. Il est protégé par un fichier *.htaccess* qui garanti que seuls les enseignants autorisés peuvent l'utiliser. Le programme de mise à jour de la base *mbraise.cgi* s'appuie sur les modules suivants :

- *braise.pm* définit la configuration de l'application (où se trouvent les différents fichiers, ...), la figure suivante est un court extrait :

```
# Ou est installée l'appli au sens du système de fichier.
$InstallDir = "/usr/local/apache/htdocs/braise";
# Ou est installée l'appli au sens HTTP.
$HTTPInstDir = "/braise";
# Le nom des fichiers et répertoires gérés.

$Exos          = "$InstallDir/donnees/Exos.xml";
$DV            = "$InstallDir/donnees/DejaVus.txt";
$HTMLDir       = "$InstallDir/donnees/HTML";
$ParamHTMLDir = "$InstallDir/donnees/ParamHTML";
$Param         = "$InstallDir/donnees/Param.xml";

# Les images utilisées pour les différentes icones.

$ImgErr = "$HTTPInstDir/images/dark.gif";
$ImgOk  = "$HTTPInstDir/images/light.gif";
$ImgChk = "$HTTPInstDir/images/check.gif";
```

- *log.pm* un petit module qui gère l'accès aux deux *logs* de l'application (mise à jour de la base d'exercices et accès à l'interface étudiant).
- *traque.pm* permet à un enseignant de suivre l'activité d'un étudiant. C'est le contenu du *log* d'accès à l'interface étudiant qui est utilisé.

Le répertoire **donnees** contient les éléments suivants :

- la base d'exercices au format XML (*Exos.xml*) pour laquelle un historique des versions est conservé (*Exos.xml.0*, *Exos.xml.1*, ...).
- l'ensemble des paramètres utiles à la constitution d'un chapitre qui sont regroupés dans un document XML (*Param.xml* cf. section 5)
- le répertoire **HTML** qui contient le résultat de la traduction en HTML des exercices. À chaque exercice correspond un répertoire dans lequel on trouve son fichier source au format L^AT_EX (*exo.tex*), le résultat de la traduction (*exo.html*), ...
- le répertoire **ParamHTML** qui reçoit la traduction en HTML des paramètres constituant les

chapitres. Il contient un répertoire par chapitre (**Algèbre**, **Les Suites**, ...) chacun d'entre eux englobant un répertoire par constituant (theme, difficultés, nature de la tâche, méthodes et connaissances).

Le fichier *Date* indique la date de dernière traduction de l'ensemble des paramètres. C'est en comparant son contenu avec la date de dernière modification du fichier *Param.xml* que le processus de traduction est automatiquement démarré lorsqu'un enseignant demande une mise à jour de la base d'exercices.

- le fichier *DejaVus.txt* qui indique quels sont les exercices déjà visionnés par un étudiant.

B Les figures



FIG. 1 – La base au format XML vue par Internet Explorer

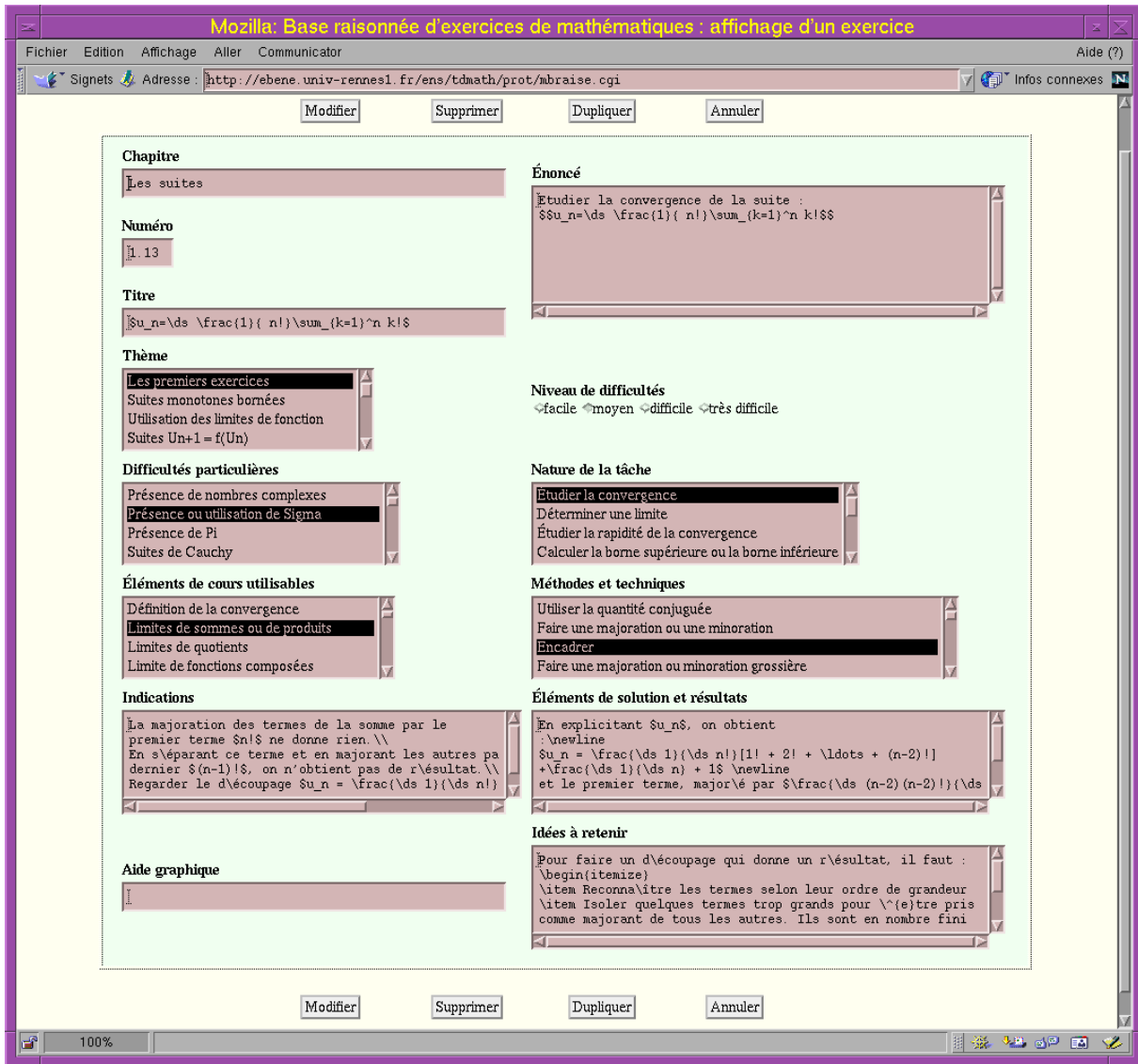


FIG. 2 – L'interface WWW de création ou modification d'exercice

Mozilla: Base raisonnée d'exercices de mathématiques : liste des exercices Les suites

Fichier Edition Affichage Aller Communicator Aide (?)

Signets Adresse : <http://ebene.univ-rennes1.fr/ens/tdmath/prot/mbraise.cgi> Infos connexes

News Downloads Software Hardware Developers Help Search Shop

Base raisonnée d'exercices de mathématiques : liste des exercices *Les suites*

Les suites Nouvel exercice Accès étudiant Suivi étudiant

Numéro	Titre	Énoncé
1.1	$u_n = \frac{n}{n^2+1}$	Etudier la suite : $u_n = \frac{n}{n^2+1}$
1.2	$u_n = \frac{n}{2n^2+1} \{ 3n^2+5 \}$	Déterminer la limite de la suite : $u_n = \frac{n}{2n^2+1} \{ 3n^2+5 \}$
1.3	$u_n = \frac{n}{2n^3+1} \{ 3n^2+5 \}$	Etudier la suite : $u_n = \frac{n}{2n^3+1} \{ 3n^2+5 \}$
1.4	$u_n = \sqrt[n]{n+1} - \sqrt[n]{n}$	Etudier la suite (on pourra préciser un N en fonction de ϵ)
1.5	$u_n = \frac{\sin(n)}{\sqrt[n]{n}}$	Etudier la suite (on pourra préciser un N en fonction de ϵ)
1.6	$u_n = \frac{\sqrt[n]{n}}{n + \sin(n)}$	Etudier la suite, définie pour $n > 1$: $u_n = \frac{\sqrt[n]{n}}{n + \sin(n)}$
1.7	$u_n = \frac{a^n - b^n}{a^n + b^n}$	Etudier la suite : $u_n = \frac{a^n - b^n}{a^n + b^n}$
1.8	$u_n = \prod_{p=2}^n \left(1 - \frac{1}{p} \right)$	En simplifiant le terme général u_n , étudier la convergence de la suite.
1.9	$u_n = \prod_{p=2}^n \left(1 + \frac{1}{p^2} \right)$	En simplifiant le terme général u_n , étudier la convergence de la suite.
1.10	$u_n = \prod_{p=2}^n \left(1 + \frac{1}{(-1)^{p+1}} \right)$	En simplifiant le terme général u_n , définir la limite de la suite.
1.11	$u_n = \sum_{k=1}^n \frac{n}{n^2+k}$	Etudier la convergence de la suite : $u_n = \sum_{k=1}^n \frac{n}{n^2+k}$
1.12	$u_n = \sum_{k=1}^n \frac{n}{\sqrt[n]{n^2+k}}$	Etudier la convergence de la suite : $u_n = \sum_{k=1}^n \frac{n}{\sqrt[n]{n^2+k}}$
1.13	$u_n = \frac{1}{n!} \sum_{k=1}^n k!$	Etudier la convergence de la suite : $u_n = \frac{1}{n!} \sum_{k=1}^n k!$
1.14	$u_n = \frac{1}{n!} \sum_{k=1}^n (n-1)k!$	Etudier la convergence de la suite : $u_n = \frac{1}{n!} \sum_{k=1}^n (n-1)k!$
1.15	$u_n = \sum_{k=0}^n \frac{1}{C_n^k}$	Montrer que la suite $u_n = \sum_{k=0}^n \frac{1}{C_n^k}$ converge vers e .
1.16	$\frac{u_{n+1}}{u_n}$ converge vers e	Soit (u_n) une suite de réels strictement positifs telle que $\frac{u_{n+1}}{u_n} \rightarrow e$. Étudier la convergence de la suite (u_n) .
2.1	$u_n = \ln \left(\frac{n+a}{1+na} \right)$	Si $a \geq 0$, déterminer la limite de la suite (u_n) .
2.2	$u_n = \arctan \left(\frac{n+a}{1+na} \right)$	Si $a \geq 0$, déterminer la limite de la suite (u_n) .
2.3	$u_n = \sqrt[n]{n}$	Etudier la convergence de la suite : $u_n = \sqrt[n]{n}$
2.4	$u_n = \sqrt[n]{\ln n}$	Etudier la convergence de la suite : $u_n = \sqrt[n]{\ln n}$
2.5	$u_n = \left(1 + \frac{a}{n} \right)^n$	Déterminer la limite de la suite : $u_n = \left(1 + \frac{a}{n} \right)^n$
2.6	$u_n = \left(\sqrt[n]{n} - 1 \right)^n$	Etudier la convergence de la suite : $u_n = \left(\sqrt[n]{n} - 1 \right)^n$
2.7	$u_{n+1} = \sqrt[n]{a^n + b^n}$	Etudier la convergence de la suite : $u_{n+1} = \sqrt[n]{a^n + b^n}$
3.1	$u_{n+1} = \sqrt{1+u_n}$	Etudier les suites définies par $u_0 \geq -1$ et $u_{n+1} = \sqrt{1+u_n}$.
3.2	$u_{n+1} = \frac{1}{\sqrt{1+u_n}}$	Etudier les suites définies par $u_{n+1} = \frac{1}{\sqrt{1+u_n}}$.
3.3	$u_{n+1} = au_n + b$	Etudier, selon les valeurs de a et b , les suites définies par $u_{n+1} = au_n + b$.
3.4	$u_n = \sqrt{2\sqrt{2\sqrt{\dots\sqrt{2}}}}$	Etudier la suite définie par $u_0 = \sqrt{2}$, $u_{n+1} = \sqrt{2u_n}$.
3.5	$u_{n+1} = 1 - \cos u_n$	Etudier la suite : $u_{n+1} = 1 - \cos u_n$ avec $u_0 \in]0, \pi[$.

FIG. 3 – La liste des exercices d'un chapitre

Les suites

Énoncé de l'exercice 1.10

En simplifiant le terme général u_n , défini pour $n \geq 2$, étudier la convergence de la suite :

$$u_n = \prod_{p=2}^n \left(1 + \frac{(-1)^{p+1}}{p} \right)$$

Thème

- [Les premiers exercices](#)

Titre

$$u_n = \prod_{p=2}^n \left(1 + \frac{(-1)^{p+1}}{p} \right)$$

Niveau de difficulté

difficile

Difficultés particulières

- [Présence de \$\Pi\$](#)

Nature de la tâche

- [Étudier la convergence](#)

Aides

- **Méthodes et techniques**
 - Regarder les rangs pairs et impairs
 - Faire un raisonnement par récurrence
- **Indications**

On peut regarder les premiers termes et donner une expression pour les termes de rang pair et les termes de rang impair.

Éléments de solutions et résultats

En explicitant les premiers termes de $u_n = \prod_{p=2}^n \frac{p + (-1)^{p+1}}{p}$, on obtient $u_n = \frac{1}{2} \cdot \frac{4}{3} \cdot \frac{3}{4} \cdot \frac{6}{5} \cdot \frac{5}{6} \cdot \frac{8}{7} \dots$, ce qui donne

l'idée de justifier par récurrence que $u_{2p} = \frac{1}{2}$, $u_{2p+1} = \frac{1}{2} \cdot \frac{2p+2}{2p+1}$.

FIG. 4 – Le résultat de la traduction en HTML d'un exercice

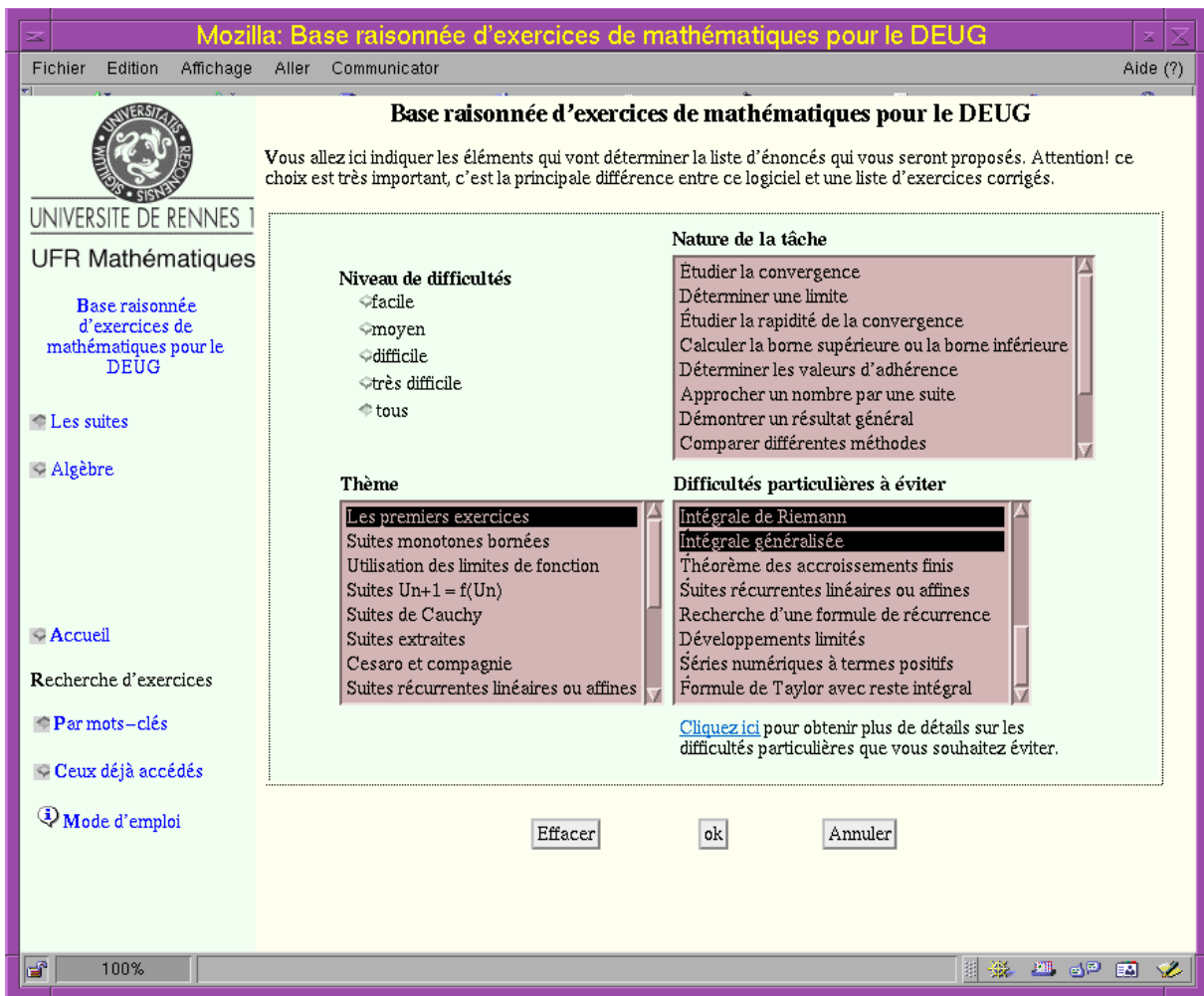



FIG. 5 – La sélection d'exercices par mots-clés

Mozilla: Base raisonnée d'exercices de mathématiques pour le DEUG

Fichier Edition Affichage Aller Communicator Aide (?)



UNIVERSITE DE RENNES 1
UFR Mathématiques

Base raisonnée d'exercices de mathématiques pour le DEUG

- Les suites
- Algèbre
- Accueil

Recherche d'exercices

- Par mots-clés
- Ceux déjà accédés

Base raisonnée d'exercices de mathématiques pour le DEUG

Résultat de votre recherche

Numéro	Titre	Niveau
6	1.6 $u_n = \frac{\sqrt{n}}{n + \sin n}$	moyen
7	1.7 $u_n = \frac{a^n - b^n}{a^n + b^n}$	moyen
8	1.8 $u_n = \prod_{p=2}^n \left(1 - \frac{1}{p}\right)$	facile
9	1.9 $u_n = \prod_{p=2}^n \left(1 - \frac{1}{p^2}\right)$	facile
10	1.10 $u_n = \prod_{p=2}^n \left(1 + \frac{(-1)^{p+1}}{p}\right)$	difficile

42 exercices correspondent à votre recherche.

Cliquez ci-dessous pour visionner les autres exercices.

← →


Modifier la recherche

100%

FIG. 6 – Le résultat d'une sélection d'exercices par mots-clés

Mozilla: Base raisonnée d'exercices de mathématiques pour le DEUG

Fichier Edition Affichage Aller Communicator Aide (?)



UNIVERSITE DE RENNES 1
UFR Mathématiques

Base raisonnée d'exercices de mathématiques pour le DEUG

- ▾ Les suites
- ▾ Algèbre
- ▾ Accueil

Recherche d'exercices

- ▾ Par mots-clés
- ▾ Ceux déjà accédés

Exercice numéro 1.10

$$u_n = \prod_{p=2}^n \left(1 + \frac{(-1)^{p+1}}{p} \right)$$

Énoncé

En simplifiant le terme général u_n , défini pour $n \leq 2$, étudier la convergence de la suite :

$$u_n = \prod_{p=2}^n \left(1 + \frac{(-1)^{p+1}}{p} \right)$$

Caractéristiques

Thème(s)

- [Les premiers exercices](#)

Niveau de difficulté : *difficile*

<p>Difficultés particulières</p> <ul style="list-style-type: none"> • Présence de \prod 	<p>Nature de la tâche</p> <ul style="list-style-type: none"> • Étudier la convergence
--	---

◆ Aides
◆ Éléments de solutions et résultats
◆ Idées à retenir

↩ [Retour au résultat de votre recherche](#)

FIG. 7 – La visualisation d'un exercice

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
- <param>
  <!-- Les niveaux de difficultés -->
  <niv>facile</niv>
  <niv>moyen</niv>
  <niv>difficile</niv>
  <niv>très difficile</niv>
  <!-- Le chapitre sur les suites -->
- <chap>
  <titre>Les suites</titre>
  - <untheme>
    <theme>Les premiers exercices|||</theme>
    <texte>en cours ...</texte>
  </untheme>
  - <untheme>
    <theme>Suites monotones bornées|||</theme>
    <texte>en cours ...</texte>
  </untheme>
  - <untheme>
    <theme>Utilisation des limites de fonction|||</theme>
    <texte>en cours ...</texte>
  </untheme>
  - <untheme>
    <theme>Suites  $U_{n+1} = f(U_n)$ |||Suites  $U_{n+1}=f(U_n)$ </theme>
    <texte>%theme 3 : Suites  $u_{n+1}=f(u_n)$  \section{Suites  $u_{n+1}=f(u_n)$ }
    %Indications pour le th\`eme\` Pour les exercices sur ce th\`eme il est
    souvent utile de repr\`e\`senter la fonction  $f$ , la droite  $y=x$ , ainsi que
    quelques termes de la suite en fonction de  $u_0$ . Ces dessins doivent permettre
    de deviner le comportement de la suite  $(u_n)$  et de voir quel type de
    m\`ethode a des chances d'\`etre efficace : \begin{enumerate} \item Soit
    d'\`eterminer d'un `candidat limite"  $\ell$  par r\`esolution de  $\boxed{f$ 
     $(\ell)=\ell$  puis majoration de  $|u_{n+1}-\ell|=|f(u_n)-f(\ell)|$  par une
    suite (en g\`eneral g\`eom\`etrique) tendant vers  $0$ . Une possibilit\`e pour
    obtenir cette majoration est d'appliquer le th\`eor\`eme des accroissements
    finis. Contrairement aux pr\`ec\`edentes, cette m\`ethode ne donne aucun
    renseignement sur la vitesse de convergence. Cette m\`ethode met en
    \`evidence le comportement de la suite selon la valeur de  $u_0$ . \` Pour
    d\`emontrer la monotonie de  $u_n$  on utilise souvent la monotonie de  $f$  : par
    exemple si  $f$  est croissante au voisinage d'un candidat limite  $\ell$ , de
     $u_0 < u_1 < \ell$  on d\`eduit par r\`ecurrence  $u_n < u_{n+1} < \ell$ . \par Les
    m\`ethodes 1 et 2 sont dites {\it m\`ethodes quantitatives}, puisqu'elles
    permettent d'obtenir des informations sur la vitesse de convergence. Pour
    exp\`erimenter ces diff\`erentes m\`ethodes, on peut commencer par l'exercice
     $u_{n+1}=\sqrt{1+u_n}$ , 3.1. %LIEN avec  $u_{n+1}=\sqrt{1+u_n}$ , 3.1)
    \end{enumerate}</texte>
  </untheme>
  - <untheme>

```

FIG. 8 – Le document XML de paramétrage d'un chapitre

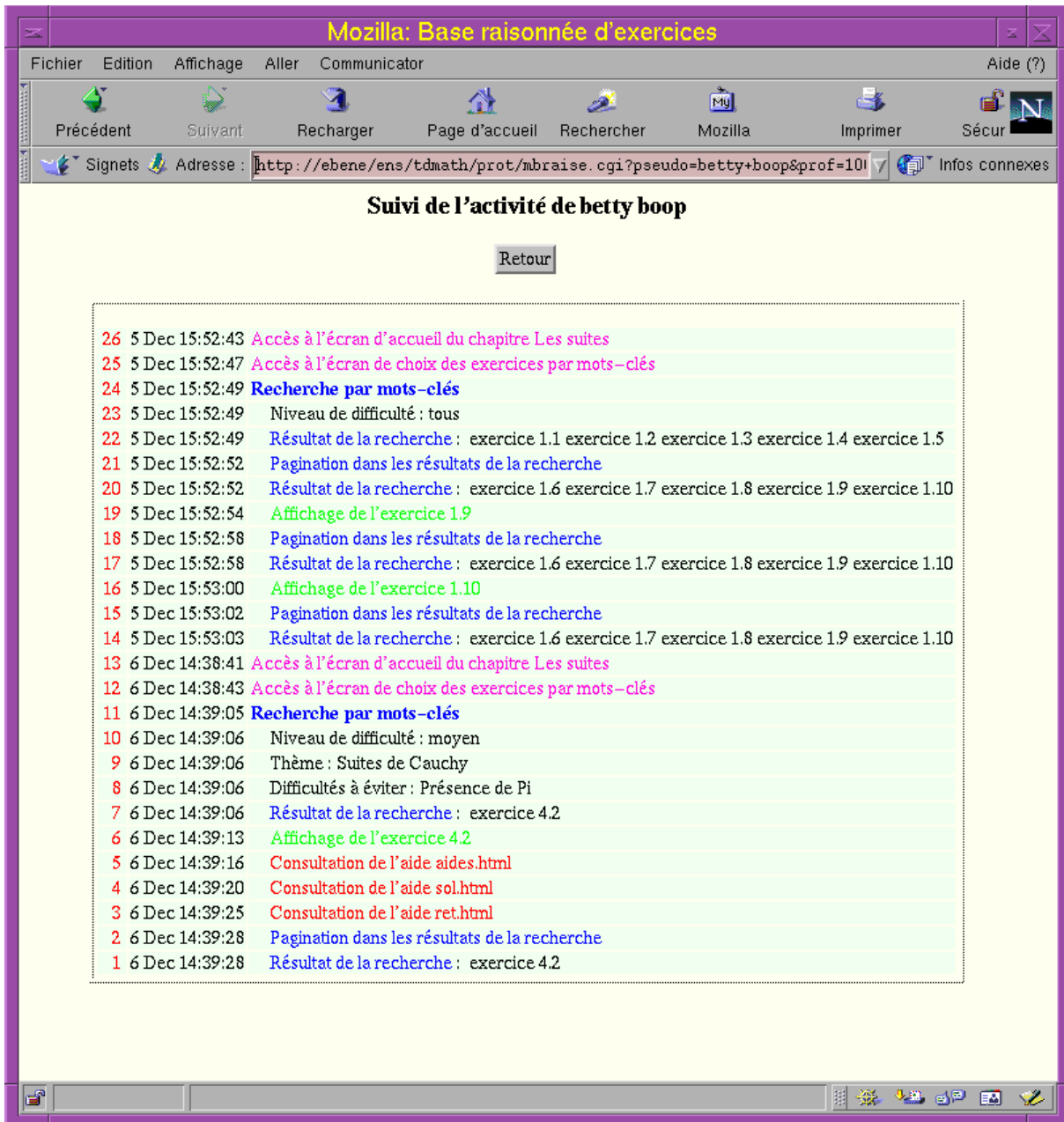


FIG. 9 – Le suivi de l'activité d'un étudiant