
DocBook et les formules mathématiques

Utiliser $\text{\LaTeX}Math$

Christophe HARO <Christophe.HARO@ac-nantes.fr>

Calixte DENIZET <Calixte.DENIZET@ac-rennes.fr>

Copyright © 2010 Christophe HARO and Calixte DENIZET

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

```
Notes de révision
dernier commit : "$Date: 2010-04-16 16:31:47 +0200 (Ven 16 avr 2010) $"
dernière compilation : 16-04-2010 16:32:07
revision : "$Revision: 16 $"
url : "$HeadURL: svn://localhost/jlatexmath/xml/doc-jlatexmath.xml $"
modifications : "$Author: haro $"
```

Résumé

Ce document montre comment utiliser quelques commandes \LaTeX pour éditer un texte scientifique au format **DocBook**. Il s'agit de réaliser une petite démonstration pour présenter le rendu obtenu dans une sortie au format PDF d'un texte Mathématique incluant des équations.

Table des matières

1. Introduction	2
2. Installation du paquetage et adaptation de la feuille de style	3
2.1. Installation de $\text{\LaTeX}Math$	3
2.2. Adaptation de la feuille de style	3
3. Exemples de composition avec $\text{\LaTeX}Math$	6
4. Complément : changer les polices de caractères par défaut	14
4.1. Télécharger l'archive des polices	15
4.2. Générer les fichiers de métriques	15
4.3. Configurer FOP	17
4.4. Utiliser les nouvelles polices	18
5. \LaTeX dans les sources de programmes	18
6. Webographie	20
7. GNU Free Documentation License	20



Source des programmes présentés

Pour des raisons techniques, les sources des programmes présentés ne sont pas exactement ceux des programmes réels. Ainsi, par exemple, lorsqu'un environnement mul-

ti-ligne de L^AT_EX nécessite de placer le caractère &, celui-ci devrait être échappé puisque c'est un caractère réservé de XML. Le code source réel, tel qu'il apparaît dans les fichiers joints, encadre les instructions des expressions mathématiques par les balises `<![CDATA[...]]>` qui échappent ainsi les caractères réservés de XML. Cependant, ces balises ne peuvent être imbriquées dans les éléments `<programlisting>`. Elles n'apparaissent donc pas dans ce document, mais il faudra veiller à les rétablir pour obtenir des programmes XML valides.

1. Introduction

Ce texte a été composé avec EMACS dans le mode majeur nXML. Le système de composition est **DocBook** en version 5.0 XML validé par le schéma RELAX NG. Les feuilles de styles XSLT sont utilisées par XMLLINT pour la validation et la résolution des inclusions externes. SAXON est le processeur XSLT et FOP réalise la composition des textes imprimés au format pdf. La composition à l'aide de FOP utilise les extensions J_LA_TE_XMath pour le rendu des équations et des expressions mathématiques.

La composition d'expressions mathématiques avec **DocBook** n'est pas simple et, surtout, donne un rendu qui n'est pas satisfaisant comparé à celui que permet L^AT_EX et ses paquetages d'extensions. Il existe différents projets qui tentent de rassembler le meilleur des deux mondes. Le projet J_LA_TE_XMath développé par Calixte DENIZET est de ceux-là. Il permet d'intégrer simplement dans un texte xml au format **DocBook** certaines des commandes L^AT_EX les plus utiles pour composer les expressions mathématiques.

J_LA_TE_XMath est une API **Java**. Son utilisation est simple, efficace et robuste. La page d'accueil du projet, à l'adresse <http://forge.scilab.org/index.php/p/jlatexmath/>, vous en dira plus.

L'objectif de cette documentation est de donner quelques exemples simples de composition d'expressions mathématiques. Ce n'est donc un tutoriel ni sur **DocBook**, ni sur L^AT_EX. Ce n'est pas non plus une étude exhaustive des possibilités de J_LA_TE_XMath. Les références en fin de texte permettront d'aller plus loin pour ceux que cela intéresse. Elle s'adresse au débutant sans expertise particulière sur **DocBook** pour lui permettre de démarrer rapidement avec la composition de textes scientifiques. Cependant, elle ne lui économisera pas les efforts nécessaires pour l'utiliser efficacement.

La Section 2, « Installation du paquetage et adaptation de la feuille de style », donne quelques indications pour installer le paquetage J_LA_TE_XMath et pour configurer la feuille de style afin d'en utiliser les fonctionnalités. La Section 3, « Exemples de composition avec J_LA_TE_XMath », montre quelques exemples de composition d'expressions. La Section 4, « Complément : changer les polices de caractères par défaut », complète ces éléments et montre comment changer les polices de caractères utilisées par défaut pour se rapprocher encore un peu plus du rendu obtenu avec L^AT_EX. La Section 5, « L^AT_EX dans les sources de programmes », montre sur un exemple simple comment insérer une expression composée avec J_LA_TE_XMath dans les commentaires du code source d'un programme informatique pour compléter sa documentation. Le document conclut par quelques références utilisées pour composer ce texte.

2. Installation du paquetage et adaptation de la feuille de style

Il n'y a pas d'installation de ℒ_AT_EX_Math au sens propre. Il suffit de placer l'archive `jlatexmath-embedded-fop-x.y.z.jar` dans un dossier accessible par **FOP** et d'ajuster la feuille de style `xsl`. Cette section indique comment faire.

2.1. Installation de ℒ_AT_EX_Math

Commencer par télécharger l'archive `jlatexmath-embedded-fop-x.y.z.zip` depuis le site du projet. L'adresse de téléchargement est <http://forge.scilab.org/index.php/p/jlatexmath/downloads/> et la version `x.y.z` est la 0.9.1(16-04-2010). Déballez cette archive dans un dossier quelconque. Elle contient le fichier `jlatexmath-embedded-fop-x.y.z.jar` qui doit être accessible à **FOP**. Vous avez deux possibilités : ajuster le `CLASSPATH` qui donne les chemins d'accès aux classes **Java** utilisées par **FOP** ou placer cette archive dans le dossier `lib/` du répertoire dans lequel est installé **FOP**.

L'ajustement de la valeur de la variable d'environnement `CLASSPATH` met en jeu des commandes qui dépendent du système d'exploitation hôte et du processeur de commandes utilisé par l'utilisateur de ce système. Sur un système UNIX par exemple et pour un utilisateur d'un tel système pour lequel le processeur de commandes est **bash** ou **sh**, l'ajustement est réalisé par la commande :

```
$ export CLASSPATH=${CLASSPATH}:chemin/jlatexmath-embedded-fop-x.y.z.jar
```

Dans cette commande, il faut évidemment remplacer `chemin` par le chemin d'accès à l'archive, c'est à dire le nom du dossier dans lequel elle a été placée. Il faut aussi remplacer `x.y.z` par leurs valeurs effectives. Le caractère `$` qui débute la ligne ne doit pas être tapé. Il est mis ici pour figurer le caractère d'invite (*prompt*) du processeur de commandes et peut différer selon la configuration du système. Ainsi, par exemple sur le calculateur sur lequel est tapé ce texte, le *prompt* est `callisto:~ haro$`.

Une méthode plus rapide consiste à placer, si c'est possible, l'archive dans le dossier `lib/` du répertoire d'installation de **FOP**. Dans ce cas, la modification de `CLASSPATH` n'est pas nécessaire.

2.2. Adaptation de la feuille de style

Il s'agit de préparer une feuille de style primaire qui va définir les commandes nécessaires pour permettre la traduction des éléments ℒ_AT_EX_Math, mais qui va aussi charger la feuille de style « normale » de la distribution **DocBook**.

Au lieu de préciser dans chaque instance de l'élément `<latex>` l'espace de nom, comme ceci :

```
<jlm:latex xmlns:jlm="http://forge.scilab.org/p/jlatexmath">
...
```

```
</jlm:latex>
```

la feuille de style définit le préfixe `jlm`. L'en-tête du fichier qui définit la feuille de style est, par exemple :

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
    xmlns:latex="http://forge.scilab.org/p/jlatexmath"
    xmlns:jlm="http://forge.scilab.org/p/jlatexmath"
    ...
    définition des autres espaces de noms
    ...
    exclude-result-prefixes="... latex jlm"
    version="1.0">

<!-- Chargement des autres feuilles de styles -->
<xsl:import href="../../../fo/docbook.xsl"/>
<xsl:import href="../../../highlighting/common.xsl" />
<xsl:import href="../../../fo/highlight.xsl" />
...
```

Bien entendu, les pointillés doivent être remplacés par ce qui manque ou éliminés si votre feuille de style est réduite aux adaptations nécessaires à J_LA_TE_XMath.

Il reste alors à rajouter la partie concernant l'élément `<latex>`.

```
...
<!-- Utiliser les commandes LaTeX pour éditer les équations -->

<xsl:template match="latex:*">
  <xsl:choose>
    <xsl:when test="@align != ''">
      <fo:block>
        <xsl:attribute name="text-align">
          <xsl:value-of select="@align"/>
        </xsl:attribute>
        <fo:instream-foreign-object>
          <xsl:copy>
            <xsl:copy-of select="@*"/>
            <xsl:if test="@align = 'center'">
              <xsl:attribute name="fwidth">
                <xsl:value-of select="concat(
                  $page.width, '-',
                  $body.start.indent, '-',
                  $page.margin.inner, '-',
                  $page.margin.outer
                )"/>
            </xsl:if>
          </xsl:copy>
        </fo:instream-foreign-object>
      </fo:block>
    </xsl:when>
    <xsl:otherwise>
```

```

    <fo:instream-foreign-object>
    <xsl:copy>
      <xsl:copy-of select="@*" />
      <xsl:apply-templates/>
    </xsl:copy>
    </fo:instream-foreign-object>
  </xsl:otherwise>
</xsl:choose>
</xsl:template>

</xsl:stylesheet>

```

Ce segment de code de la feuille de style compose le fichier `latex.xsl` placé dans le dossier `exemples/` de l'archive du projet. Il suffit donc de la copier/coller. Finalement, l'adaptation complète de la feuille de style devient :

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
  xmlns:latex="http://forge.scilab.org/p/jlatexmath"
  xmlns:jlm="http://forge.scilab.org/p/jlatexmath"
  ...
  définition des autres espaces de noms
  ...
  exclude-result-prefixes="... latex jlm"
  version="1.0">

  <!-- Chargement des autres feuilles de styles -->
  <xsl:import href="../../../fo/docbook.xsl"/>
  <xsl:import href="../../../highlighting/common.xsl" />
  <xsl:import href="../../../fo/highlight.xsl" />

  ...

  <!-- Utiliser les commandes LaTeX pour éditer les équations -->

  <xsl:template match="latex:*">
    <xsl:choose>
      <xsl:when test="@align != ''">
        <fo:block>
          <xsl:attribute name="text-align">
            <xsl:value-of select="@align"/>
          </xsl:attribute>
          <fo:instream-foreign-object>
            <xsl:copy>
              <xsl:copy-of select="@*" />
              <xsl:if test="@align = 'center'">
                <xsl:attribute name="fwidth">
                  <xsl:value-of select="concat(
                    $page.width, '-',
                    $body.start.indent, '-',
                    $page.margin.inner, '-',
                    $page.margin.outer
                  )"/>
                </xsl:attribute>
              </xsl:if>
              <xsl:apply-templates/>
            </xsl:copy>
          </fo:instream-foreign-object>
        </fo:block>
      </xsl:when>

```

```
<xsl:otherwise>
  <fo:instream-foreign-object>
<xsl:copy>
  <xsl:copy-of select="@*" />
  <xsl:apply-templates/>
</xsl:copy>
  </fo:instream-foreign-object>
</xsl:otherwise>
</xsl:choose>
</xsl:template>

</xsl:stylesheet>
```

Pour éditer un texte, il suffit de définir les espaces de noms et de préfixer les éléments dans le texte. Les préfixes définis sont libres. Ce texte, par exemple, utilise le préfixe `jlm:` pour les éléments `<latex>`, mais ce n'est pas une obligation. Les définitions utilisées pour ce texte sont les suivantes :

```
<article xml:id="doc-jlatexmath"
  xmlns="http://docbook.org/ns/docbook"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xi="http://www.w3.org/2001/XInclude"
  xmlns:svg="http://www.w3.org/2000/svg"
  xmlns:mml="http://www.w3.org/1998/Math/MathML"
  xmlns:html="http://www.w3.org/1999/xhtml"
  xmlns:db5="http://docbook.org/ns/docbook"
  xmlns:jlm="http://forge.scilab.org/p/jlatexmath"
  xml:lang="fr"
  version="5.0">
...
</article>
```

Ce qui permet ensuite de composer les éléments comme cela a été précisé au tout début de cette section.

3. Exemples de composition avec J^LA^TE_XMath

Exemple 1. Une première tentative

Commençons par une première tentative. Il s'agit d'une équation simple, sans titre ni numérotation. Le programme source pour cette équation est :

```
<para>
  <jlm:latex>
    \sum_{i=0}^{i=n}{i} = \frac{n \times (n+1)}{2}
  </jlm:latex>
</para>
```

Il suffit donc de placer les instructions `LATEX` qui composent l'équation dans un paragraphe. on obtient :

$$\sum_{i=0}^{i=n} i = \frac{n \times (n+1)}{2}$$

La même expression peut aussi être composée pour être intégrée au texte de la ligne en cours. Il s'agit donc d'une expression « en ligne » (*in-line*). Il suffit de définir l'attribut `style` de l'élément `<jlm:latex>` et de lui attribuer la valeur "text", comme ceci :

```
<para>
  Ceci est le texte d'un paragraphe dont une ligne contient une expression
  mathématique composée avec <jlm:latex style="text">\LaTeX</jlm:latex>.
  L'expression est

  <jlm:latex style="text">
    \sum_{i=0}^{i=n}{i} = \frac{n \times (n+1)}{2}
  </jlm:latex>

  et elle s'intègre parfaitement au texte. La taille des caractères est ajustée
  automatiquement.
</para>
```

On obtient alors le résultat suivant :

Ceci est le texte d'un paragraphe dont une ligne contient une expression mathématique composée avec L^AT_EX. L'expression est $\sum_{i=0}^{i=n} i = \frac{n \times (n+1)}{2}$ et elle s'intègre parfaitement au texte. La taille des caractères est ajustée automatiquement.

La formule n'est pas vraiment impressionnante et pas difficile à préparer, même sans J^AT_EX^Math. Voyons un second exemple, toujours sans numérotation, mais un peu plus compliqué.

Fin de l'Exemple 1, « Une première tentative ».

Exemple 2. Un exemple d'équation plus élaborée

Voici le résultat à obtenir :

$$\phi_n(\kappa) = \frac{1}{4\pi^2\kappa^2} \int_0^\infty \frac{\sin(\kappa R)}{\kappa R} \frac{\partial}{\partial R} \left[R^2 \frac{\partial D_n(R)}{\partial R} \right] dR$$

Les instructions nécessaires pour obtenir ce résultat sont évidemment plus difficiles à écrire. Le programme est le suivant :

```
<para>
  <jlm:latex>
    \phi_n(\kappa) = \frac{1}{4\pi^2\kappa^2} \int_0^\infty \frac{\sin(\kappa R)}{\kappa R}
    \frac{\partial}{\partial R} \left[ R^2 \frac{\partial D_n(R)}{\partial R} \right] dR
  </jlm:latex>
</para>
```

Essayons encore avec une expression utilisant une matrice. Voici ce que nous souhaitons obtenir :

$$\det \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{n1} & \cdots & \cdots & a_{nn} \end{bmatrix} \stackrel{\text{def}}{=} \sum_{\sigma \in \mathfrak{S}_n} \varepsilon(\sigma) \prod_{k=1}^n a_{k\sigma(k)}$$

Le programme s'écrit :

```
<para>
  <jlm:latex>
    \det
    \begin{bmatrix}
      a_{11}&a_{12}&\&\cdots&a_{1n}\\
      a_{21}&\&\ddots&\&\vdots\\
      \vdots&\&\ddots&\&\vdots\\
      a_{n1}&\&\cdots&\&a_{nn}
    \end{bmatrix}
    \overset{\{\mathrm{def}\}}{=}\sum_{\sigma\in\mathfrak{S}_n}\varepsilon(\sigma)\prod_{k=1}^n
    a_{k\sigma(k)}
  </jlm:latex>
</para>
```

Remarquer l'utilisation de la police de caractères `\mathfrak`. Bien entendu, nous devrions utiliser des environnements plus spécialisés pour améliorer la composition, par exemple pour ajuster l'espacement qui précède et qui suit le signe $\stackrel{\text{def}}{=}$. C'est tout à fait possible avec les commandes `\mathclap`, `\mathllap`, ... qui sont reconnues par J^AT_EX^Math mais nous en resterons là pour cette initiation.

Fin de l'Exemple 2, « Un exemple d'équation plus élaborée ».

Exemple 3. Équation numérotée avec son titre

Les équations sont habituellement numérotées afin de pouvoir y faire référence dans le texte. Pour que l'équation soit numérotée, elle doit être enveloppée dans un élément `<equation>` et un titre doit être défini dans le sous-élément `<title>`. Enfin, l'équation sera référencée à l'aide de l'identifiant défini comme valeur de l'attribut `id` de l'élément `<equation>`. Le programme de l'équation précédente devient :

```
<para>
  <equation xml:id="matrice">
    <title>Une équation numérotée</title>

    <jlm:latex>
      \det
      \begin{bmatrix}
        a_{11}&a_{12}&\&\cdots&a_{1n}\\
        a_{21}&\&\ddots&\&\vdots\\
        \vdots&\&\ddots&\&\vdots\\
        a_{n1}&\&\cdots&\&a_{nn}
      \end{bmatrix}
      \overset{\{\mathrm{def}\}}{=}\sum_{\sigma\in\mathfrak{S}_n}\varepsilon(\sigma)\prod_{k=1}^n
      a_{k\sigma(k)}
    </jlm:latex>
  </equation>
</para>
```

```
a_{k\sigma(k)}
</jlm:latex>

</equation>
</para>
```

Avec le résultat suivant :

$$\det \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{n1} & \cdots & \cdots & a_{nn} \end{bmatrix} \stackrel{\text{def}}{=} \sum_{\sigma \in \mathfrak{S}_n} \varepsilon(\sigma) \prod_{k=1}^n a_{k\sigma(k)} \quad (1)$$

La référence à cette équation utilise son identifiant dans le texte, comme ceci :

L'équation (1) ci-dessus définit le déterminant de la matrice. Elle utilise la police `\mathfrak`.

Le programme de la référence est :

```
<para>
  L'équation <xref xlink:href="#matrice"/> ci-dessus définit le déterminant
  de la matrice. Elle utilise la police \mathfrak.
</para>
```

Remarquer que le numéro dans le texte est « cliquable ». En fait, il est programmé dans le texte comme un lien sur l'identifiant de l'équation.

Lorsque l'équation est numérotée, l'espace de la largeur de la page est divisé en deux parties. La partie gauche reçoit l'expression et la partie droite le numéro de référence. Par défaut, l'expression est centrée dans sa partie et le numéro est cadré à droite dans la sienne. Ces placements sont configurables en ajustant la feuille de style `xs1`, mais ces ajustements dépassent le cadre de cette documentation d'initiation. Les références en fin d'article permettent d'aller plus loin.

Fin de l'Exemple 3, « Équation numérotée avec son titre ».

Exemple 4. Équation à « plusieurs dimensions »

L'expression « plusieurs dimensions » signifie simplement que l'équation occupera plusieurs lignes dont il faut assurer une présentation harmonieuse. C'est ce que la culture L^AT_EX désigne par l'expression « environnement *multi-line* ». Voyons d'abord une équation à deux dimensions.

La suite de Fibonacci est définie par :

$$f_i = \begin{cases} 1 & \text{si } i = 1 \text{ ou } i = 2, \\ f_{i-1} + f_{i-2} & \text{si } i > 2. \end{cases} \quad (2)$$

La suite (2) est définie par une expression présentée sur deux lignes selon le domaine des valeurs de i . Pour obtenir ce résultat, le programme utilise l'environnement `cases` de L^AT_EX. Comme l'équation devient compliquée, elle est composée dans un fichier séparé dont le nom est ici `eq-2dim.xml`. Voici d'abord le contenu significatif de ce fichier :

```
<jlm:latex xmlns:jlm="http://forge.scilab.org/p/jlatexmath">
  f_i =
  \begin{cases}
    1 & \text{si } i = 1 \text{ ou } i = 2, \\
    f_{i-1} + f_{i-2} & \text{si } i > 2.
  \end{cases}
</jlm:latex>
```

Le texte du paragraphe ci-dessus qui définit la suite est alors :

```
<para>
  La suite de Fibonacci est définie par :

  <equation xml:id="fibonacci">
    <title>Définition de la suite de Fibonacci</title>

    <xi:include
      xmlns:xi="http://www.w3.org/2001/XInclude"
      href="equations;eq-2dim.xml"
      parse="xml"/>
    </equation>
  </para>
```

Ce programme utilise les entités ` `, qui est l'entité permettant d'insérer une espace typographique insécable dans le texte et `&projet;`, dont la valeur est le chemin d'accès au fichier `eq-2dim.xml` dans le système de gestion de fichiers du calculateur hôte. Voyez les références en fin d'article pour savoir comment définir et charger ces entités. Voyez aussi l'en-tête du code source de cet article pour apprendre comment cela a été fait ici.



Fichiers de l'exemple

Le fichier qui contient le code source de cet article est `doc-jlatexmath.xml`. Il est placé dans le dossier `xml` du répertoire de la documentation.

L'équation de l'exemple est définie dans le fichier `eq-2dim.xml` du dossier `equations`. Étudiez ce fichier car il est légèrement différent de ce qui est présenté ci-dessus. C'est la version de l'archive de documentation qui doit être utilisée.

Une difficulté plus contraignante est l'élément `<include>` qui permet de charger le contenu du fichier désigné pour l'inclure dans le texte. C'est que les processeurs XSLT, comme **saxon** ou **xsltproc** par exemple, ne savent pas interpréter l'élément `<include>` pour résoudre les références et inclure des fichiers externes. La solution consiste à réaliser une compilation du texte en plusieurs passes. La première passe utilise le processeur **xmllint** (noter les deux caractères 'l') qui résout les références externes et compose un fichier `xml` intermédiaire dans lequel le fichier externe est chargé à sa place. Les autres passes compilent le texte du fichier intermédiaire. Voyez les références en fin d'article pour plus de détails sur l'inclusion de fichiers dans un texte composé pour **DocBook**.

Fin de l'Exemple 4, « Équation à « plusieurs dimensions » ».

Exemple 5. Une équation vraiment compliquée

Cette fois, voici un texte mathématique vraiment compliqué. L'expression se présente ainsi :

$$\begin{cases} \dot{x}(t) = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & & \ddots & \ddots & \\ 0 & & & \ddots & 0 \\ \hat{a}_k & & & 0 & 1 \\ -a_0 & -a_1 & \dots & \dots & -a_{n-1} \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 1 \end{pmatrix} e(t) \\ s(t) = (b_0 \ b_1 \ \dots \ b_m \ 0 \ \dots \ 0)x(t) \end{cases} \quad (3)$$

avec :

$$\hat{a}_k = \underbrace{\left[\sum_{i=1}^k \frac{x_i x_i^T}{\sigma_i^2} \right]^{-1} \left[\sum_{i=1}^k \frac{x_i y_i}{\sigma_i} \right]}_{\text{D'après la règle énoncée plus haut}} \quad \forall k, \text{ tel que } 1 \leq k < N$$

C'est encore une équation définie dans un fichier xml séparé. Il est inclus par les instructions suivantes :

```
<xi:include
  xmlns:xi="http://www.w3.org/2001/XInclude"
  href="equations;eq-tres-compliquee.xml"/>
```

Le texte de ce fichier est le suivant :

```
<?xml version="1.0" encoding="UTF-8" ?>

<para
  xmlns="http://docbook.org/ns/docbook" version="5.0"
  xmlns:jlm="http://forge.scilab.org/p/jlatexmath"
  xml:lang="fr">

  <equation xml:id="eq-tres-compliquee">
    <title>Édition d'une équation complexe</title>

    <jlm:latex>
      \left\{
        \begin{array}{l}
          \dot{x}(t)=

          \left(
            \begin{array}{ccccc}
              0 & & 1 & & 0 \\
              \vdots & & & \ddots & \\
              0 & & & \ddots & 0 \\
              \underline{\hat{a}}_k & & & 0 & 1 \\
              -a_0 & -a_1 & \dots & \dots & -a_{n-1}
            \end{array}
          \right)

          x(t) +


```

```

\left(
\begin{array}{c}
0 \\
\vdots \\
0 \\
\vdots \\
1
\end{array}
\right)
e(t)

\\

s(t) =
\left(
\begin{array}{cccccc}
b_0 & b_1 & \ldots & b_m & 0 & \ldots & 0
\end{array}
\right)
x(t)
\end{array}
\right.
.
</jlm:latex>

<?linebreak?>
avec\&#xA0;:
<?linebreak?>

<jlm:latex>
\underbrace{
\displaystyle
\frac{\sum_{i=1}^k \underline{x}_i \underline{x}_i^T}{\sum_{i=1}^k \underline{x}_i y_i}
}_{\text{D'après la règle énoncée plus haut}}
\hspace{1cm} \text{for all } k, \text{ tel que } 1 \leq k < N
</jlm:latex>
</equation>
</para>

```

Le texte placé en indice, sous l'accolade, peut aussi être programmé en utilisant la commande `\text`, comme ceci :

```

...}_{\text{D'après la règle énoncée plus haut}}\hspace etc...

```

Mais alors, le rendu n'est plus le même.

Fin de l'Exemple 5, « Une équation vraiment compliquée ».

Exemple 6. Un autre exemple non trivial

Voici encore une dernière formule compliquée :

$$\forall f \in C^\infty\left(\left[-\frac{T}{2}; \frac{T}{2}\right]\right), \forall t \in \left[-\frac{T}{2}; \frac{T}{2}\right], f(\tau) = \sum_{k=-\infty}^{+\infty} e^{2i\pi \frac{k}{T}t} \times \underbrace{\frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) e^{-2i\pi \frac{k}{T}t} dt}_{a_k = \tilde{f}(\nu = \frac{k}{T})} \quad (4)$$

Elle est aussi définie dans un fichier indépendant qui est inclus ici par les instructions :

```
<para>
  Voici encore une dernière formule compliquée&nbsp;:

  <xi:include
    xmlns:xi="http://www.w3.org/2001/XInclude"
    href="equations;eq-tres-compliquee-2.xml"/>
</para>
```

Le texte du fichier de définition de la formule est :

```
<?xml version="1.0" encoding="UTF-8" ?>

<para
  xmlns="http://docbook.org/ns/docbook" version="5.0"
  xmlns:jlm="http://forge.scilab.org/p/jlatexmath"
  xml:lang="fr">

  <equation xml:id="eq-tres-compliquee-2">
    <title>Nouvelle équation très compliquée</title>

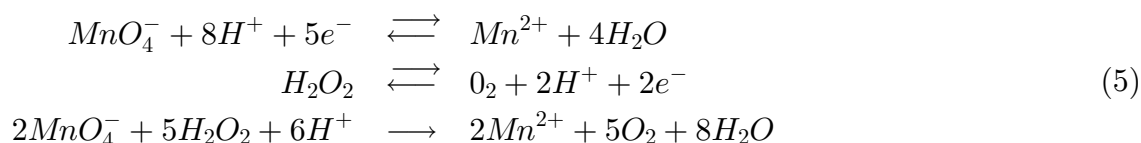
    <jlm:latex>
      \forall f \in C^\infty
      \left(
        \left[
          -\frac{T}{2} ; \frac{T}{2}
        \right]
      \right)
      \left(
        \forall t \in \left[ -\frac{T}{2} ; \frac{T}{2} \right],
        f(\tau) = \sum_{k = -\infty}^{+\infty} e^{2i\pi \frac{k}{T}t} \times
        \underbrace{
          \frac{1}{T}
          \int_{-\frac{T}{2}}^{\frac{T}{2}}
          f(t) e^{-2i\pi \frac{k}{T}t} dt
        }_{a_k = \tilde{f}(\nu = \frac{k}{T})}
      \right)
    </jlm:latex>
  </equation>
</para>
```

Là encore et comme pour l'équation (1), il faudrait faire quelques efforts supplémentaires pour améliorer la composition. Restons-en là.

Fin de l'Exemple 6, « Un autre exemple non trivial ».

Exemple 7. Édition d'une formule chimique

Le dernier exemple montre comment éditer une formule chimique.



Il semble inutile de revenir sur les instructions de chargement du fichier qui définit l'équation. Le texte de ce fichier est le suivant :

```
<?xml version="1.0" encoding="UTF-8" ?>

<para
  xmlns="http://docbook.org/ns/docbook" version="5.0"
  xmlns:jlms="http://forge.scilab.org/p/jlmsmath"
  xml:lang="fr">

  <equation xml:id="eq-chimie">
    <title>Édition d'une équation en chimie</title>

    <jlms:latex>
      \begin{eqnarray}
        \text{MnO}_4^- + 8\text{H}^+ + 5\text{e}^- & & \stackrel{\displaystyle}{\longrightarrow} & \stackrel{\displaystyle}{\longleftarrow} & \text{Mn}^{2+} + 4\text{H}_2\text{O} \\
        \text{H}_2\text{O}_2 & & \stackrel{\displaystyle}{\longrightarrow} & \stackrel{\displaystyle}{\longleftarrow} & \text{O}_2 + 2\text{H}^+ + 2\text{e}^- \\
        2\text{MnO}_4^- + 5\text{H}_2\text{O}_2 + 6\text{H}^+ & \longrightarrow & 2\text{Mn}^{2+} + 5\text{O}_2 + 8\text{H}_2\text{O}
      \end{eqnarray}
    </jlms:latex>
  </equation>
</para>
```

La dernière ligne devrait être précédée d'une ligne horizontale programmée avec la commande `\hline`, mais celle-ci ne semble pas fonctionner¹. Ici encore, des environnements spécialisés de \LaTeX permettent de composer des formules chimiques de qualité. Ils sont utilisables avec $\text{\LaTeX}_{\text{Math}}$, mais leur utilisation dépasse le cadre de cette présentation.

Fin de l'Exemple 7, « Édition d'une formule chimique ».

4. Complément : changer les polices de caractères par défaut

Cette section n'expose pas une fonctionnalité particulière de $\text{\LaTeX}_{\text{Math}}$, bien que l'archive intègre tout ce qui est nécessaire. On souhaite obtenir un rendu global du fichier pdf généré par **FOP**

¹ pour l'instant(16-04-2010). La situation doit évoluer et le problème sera probablement corrigé lorsque vous lirez ces lignes.

aussi proche que possible de celui obtenu avec du « pur » L^AT_EX. L'objectif est ambitieux, aussi allons nous nous limiter, pour l'instant, à préparer l'environnement et la feuille de style pour charger et utiliser des polices de caractères de la famille *Computer Modern (CM)* en remplacement de celles utilisées par défaut avec **DocBook**. Ce n'est qu'une première tentative, de portée limitée, afin de rester simple et de permettre à tout débutant de réussir cette évolution. Comme cette évolution est tout à fait possible indépendamment de ℒ_AT_EX_{Math}, la suite de cette section ne propose pas d'utiliser les polices de caractères embarquées dans l'archive du projet. Elle montre comment installer, configurer et utiliser ces polices indépendamment.

L'évolution est réalisée en quatre étapes.

1. Récupérer dans un dossier quelconque les fichiers de polices de caractères au format `ttf` depuis l'adresse <http://sourceforge.net/projects/cm-unicode/files/cm-unicode/0.7.0/cm-unicode-0.7.0-ttf.tar.xz> ;
2. générer les fichiers de métriques des fonts à utiliser comme indiqué à l'adresse <http://www.sagehill.net/docbookxsl/AddFont.html> ou à l'adresse <http://xmlgraphics.apache.org/fop/0.95/fonts.html> ;
3. modifier le fichier de configuration de **FOP** pour définir les nouvelles polices et leurs métriques et associer les fichiers correspondants aux noms des polices ;
4. adapter la feuille de style pour charger et utiliser effectivement ces polices.

Passer ces quatre étapes avec succès n'est pas difficile, mais il faut être attentif et soigneux.

4.1. Télécharger l'archive des polices

La première étape est élémentaire. Il s'agit de télécharger l'archive et de la déballer dans un dossier quelconque. Dans la suite, ce dossier est désigné par `FontS`. Par conséquent, remplacer partout ce nom par le chemin d'accès complet.

4.2. Générer les fichiers de métriques

Générer les fichiers de métriques est un peu plus délicat. Cet exemple va supposer que nous voulons installer la police de caractères CMUSerif-Roman. C'est celle qui est utilisée dans le texte que vous lisez. Le fichier qui contient cette font est `FontS/cmunrm.ttf`. Pour déterminer à quelle police correspond un fichier du dossier `FontS`, il suffit de consulter le fichier texte `FontS/Fontmap.CMU`. Pour observer les glyphes correspondantes, il suffit d'ouvrir le fichier avec le lecteur adapté.

La distribution de **FOP** contient la classe qui permet de générer les métriques. Soit `FOP` le dossier d'installation. Ce dossier contient les sous-dossiers `FOP/build/` et `FOP/lib/`. On commence par faire de ce dernier répertoire le répertoire de travail :

```
callisto:~ haro$ cd FontS/lib
callisto:lib haro$
```

Tout ce qui précède le premier caractère \$ est l'invite du processeur de commandes et ne doit donc pas être tapé. Le second exemplaire de ce caractère est le caractère de déréréfencement de la variable d'environnement qui le suit et, lui, doit être tapé si vous avez défini cette variable. Sinon, vous devez taper à la place de `${FOP}` le chemin d'accès complet au dossier d'installation de la distribution de **FOP** que vous utilisez. Ainsi, par exemple sur le PC que j'utilise pour saisir ce texte, **FOP** est installé dans le dossier `fop-0.95/` de mon répertoire de travail. Je dois donc taper, après l'invite du processeur de commandes :

```
cd ../fop-0.95/lib
```

Il suffit alors de lancer la commande de génération des métriques, comme ceci :

```
callisto:lib haro$ java -cp \
    ../build/fop.jar:avalon-framework-4.2.0.jar:\
    commons-logging-1.0.4.jar:\
    commons-io-1.3.1.jar:\
    xmlgraphics-commons-1.3.1.jar \
    org.apache.fop.fonts.apps.TTFReader \
    ${FONTS}/cmunrm.ttf \
    ${FONTS}/cmunrm.xml
```

Le caractère `\` en bout de ligne est le caractère de suite, qui indique au processeur de commandes que la commande n'est pas terminée et qu'elle se poursuit sur la ligne suivante. Cette facilité est utilisée ici pour améliorer la présentation du texte, mais on peut normalement taper toute la commande sans se préoccuper des retours à la ligne. Bien entendu, il faut remplacer les indications de versions des paquetages par les valeurs de ceux que vous avez effectivement installés avec **FOP**.

Remarquer que la commande ci-dessus n'est pas la même que celle indiquée à l'adresse <http://xmlgraphics.apache.org/fop/0.95/fonts.html#Base-14+Fonts>. L'archive `xmlgraphics-commons-1.3.1.jar` n'est pas indiquée dans la commande du site.

Lorsque tout se passe bien, le système répond :

```
TTF Reader for Apache FOP 0.95

Parsing font...
Reading /Users/haro/fop-0.95/fonts-Latex/cm-unicode.ttf/cmunrm.ttf...
Font Family: [CMU Serif]
Creating xml font file...
Creating CID encoded metrics...
Writing xml font file /Users/haro/fop-0.95/fonts-Latex/cm-unicode.ttf/cmunrm.xml...
This font contains no embedding license restrictions.

XML font metrics file successfully created.
callisto:lib haro$
```

Le dossier `${FONTS}` contient alors le nouveau fichier `cmunrm.xml` des métriques. Cette génération doit être reprise pour chacune des polices à installer.

Les métriques étant toutes générées, il faut à présent indiquer à **FOP** où trouver les fichiers de polices et les métriques correspondantes.

4.3. Configurer FOP

Il s'agit de modifier le fichier `#{FOP}/conf/fop.xconf`. Commencer par en faire une copie de sauvegarde. La copie étant réalisée, le fichier est complété ainsi :

```
...
<!-- Information for specific renderers -->
<!-- Uses renderer mime type for renderers -->
<renderers>
  <renderer mime="application/pdf">
    ...
    <font>
      ...

      <!-- DÉBUT DES COMPLÉMENTS NECESSAIRES AUX NOUVELLES POLICES -->
      <font
        metrics-url="file:///Users/haro/fop-0.95/fonts-LaTeX/cm-unicode.ttf/cmunrm.xml"
        kerning="yes"
        embed-url="file:///Users/haro/fop-0.95/fonts-LaTeX/cm-unicode.ttf/cmunrm.ttf">
        <font-triplet name="CMUSerif-Roman" style="normal" weight="normal"/>
      </font>

      <font
        metrics-url="file:///Users/haro/fop-0.95/fonts-LaTeX/cm-unicode.ttf/cmunbx.xml"
        kerning="yes"
        embed-url="file:///Users/haro/fop-0.95/fonts-LaTeX/cm-unicode.ttf/cmunbx.ttf">
        <font-triplet name="CMUSerif-Roman" style="normal" weight="bold"/>
      </font>

      <font
        metrics-url="file:///Users/haro/fop-0.95/fonts-LaTeX/cm-unicode.ttf/cmunti.xml"
        kerning="yes"
        embed-url="file:///Users/haro/fop-0.95/fonts-LaTeX/cm-unicode.ttf/cmunti.ttf">
        <font-triplet name="CMUSerif-Roman" style="italic" weight="normal"/>
      </font>
      <!-- FIN DES COMPLÉMENTS NECESSAIRES AUX NOUVELLES POLICES -->

      <directory>/Library/Fonts</directory>
      <auto-detect/>
    </font>

    <!-- This option lets you specify additional options on an XML handler -->
    <!--xml-handler namespace="http://www.w3.org/2000/svg">
      <stroke-text>false</stroke-text>
    </xml-handler-->

  </renderer>
  ...
```

Bien entendu, il faut remplacer les chemins d'accès des url à leurs valeurs effectives dans le SGF hôte. Attention aussi à bien préfixer chaque URL par `file:///` avec TROIS caractères '/'. Les attributs de l'élément `` sont commentés à l'adresse <http://xmlgraphics.apache.org/fop/0.95/fonts.html>. L'attribut `embed-url` permet d'embarquer dans le pdf généré les fonts correspondantes, ce qui permet au visualiseur d'afficher le document, même si la machine sur lequel il est affiché ne dispose pas des polices.

4.4. Utiliser les nouvelles polices

Il reste à modifier la feuille de style primaire pour utiliser les nouvelles polices. Pour cela, on ajoute à la feuille de style déjà préparée dans les sections précédentes les éléments suivants :

```
<!-- Polices de caractères à utiliser -->
<!-- ... pour le document -->
<xsl:param name="title.font.family">Times New Roman</xsl:param>
<xsl:param name="body.font.family">CMU Serif-Roman</xsl:param>
<xsl:param name="body.font.size">12pt</xsl:param>
<!-- ... pour les listings de programmes -->
<xsl:attribute-set name="monospace.verbatim.properties">
  <xsl:attribute name="font-family">Courier</xsl:attribute>
  <xsl:attribute name="font-size">9pt</xsl:attribute>
</xsl:attribute-set>
```

Ce sont exactement les paramètres utilisés pour préparer le document que vous lisez. Cependant, il est aussi possible de s'en tenir aux polices installées par défaut avec J_LA_TE_XMath. Voyez pour cela le fichier `examples/latex.xml` dans l'archive du projet. Vous obtiendrez les mêmes résultats avec moins d'efforts. Encore une fois, la procédure décrite ci-dessus est indépendante de J_LA_TE_XMath et permet d'installer une police quelconque.

Il faudra, dorénavant, lancer **FOP** avec l'option `-c`, complétée du chemin d'accès au nouveau fichier de configuration, comme ceci par exemple :

```
callisto:xml haro$ fop -c ${FOP}/conf/fop.xconf \
                        ${FO}/doc-jlatexmath.fo
                        -pdf ${JLATEXMATH}/imprimables/doc-jlatexmath.pdf
```

`${FOP}` est le dossier d'installation de **FOP**, `${FO}` est le dossier qui reçoit le fichier intermédiaire au format `fo` à convertir. `${JLATEXMATH}` est le dossier du projet actuellement en développement. Dans cet exemple, il s'agit tout simplement de la documentation que vous lisez.

Il ne vous reste plus qu'à expérimenter vos propres configurations. Les références précédentes et celles de la Section 6, « Webographie », devraient vous y aider.

5. L_AT_EX dans les sources de programmes

Lorsque le document expose des sources de programmes informatiques, il peut être intéressant d'embarquer, dans les commentaires du programme, des expressions mathématiques composées avec J_LA_TE_XMath. Cette section montre un exemple simple de documentation embarquée dans le code source d'un programme.

Le code source d'un programme informatique est inséré dans un document au format **DocBook** à l'aide de l'élément `<programlisting>`. L'attribut `language` de cet élément permet de préciser le langage informatique dans lequel est écrit le code du programme. Des **API** complémentaires autorisent, par exemple, la coloration syntaxique du code. Les lecteurs intéressés trouveront tous les

renseignements nécessaires à l'adresse <http://www.sagehill.net/docbookxsl/ProgramListings.html> ainsi qu'à l'adresse <http://xmlguru.cz/2006/07/docbook-syntax-highlighting>.

Voici un exemple de présentation de code source. Il s'agit d'un extrait d'une classe écrite en langage Eiffel.

```
indexing
  description :
    "{
      Le terme de rang `n' de la suite de Fibonacci (Version récursive
      terminale).
      La suite de Fibonacci est définie par :
        
$$f_i = \begin{cases} 1 & \text{si } i = 1 \text{ ou } i = 2, \\ f_{i-1} + f_{i-2} & \text{si } i > 2. \end{cases}$$

    }"
  auteurs      : "Christophe HARO"
  copyright_auteur : "Auteur : (c) Christophe HARO, 2010"
  licenses_url  : "http://www.gnu.org/licences/gpl-3-0.html"
  licence      : "GPL"
  projet       : "JLATEXMATH"
  dossier      : "${DOCBOOK}/jlatexmath/"
  fichier      : "fibonacci.e"
  date         : "$Date: 2010-04-16 16:31:47 +0200 (Ven 16 avr 2010) $"
  revision     : "$Revision: 16 $"
  url          : "$HeadURL: svn://localhost/jlatexmath/xml/doc-jlatexmath.xml $"
  modifications : "$Author: haro $"

...

feature -- Requête publique

  fibonacci(n : INTEGER) : INTEGER is
    -- Le terme de rang `n'
    require
      n > 0

    do
      Result := fibo(n, 1, 1)

    ensure
      "{ Result = le terme de rang `n' }"
    end

feature {} -- Requête privée

  fibo(n, F1, Resultat : INTEGER) : INTEGER is
    -- Le terme de rang `n'
    do
      if n < 3 then
        Result := Resultat
      else
        Result := fibo(n-1, Resultat, Resultat + F1)
      end
    end

  end

...
```

Remarquer quelques défauts dans ce texte. D'une part, la coloration syntaxique est perdue après l'inclusion dans le commentaire d'en-tête. D'autre part, il n'est pas possible de réaliser une double inclusion. Il serait ainsi intéressant de pouvoir programmer :

```
<xi:include
  xmlns:xi="http://www.w3.org/2001/XInclude"
  href="programmes;fibonacci"
  parse="text"/>
```

Le programme source serait chargé et chargerait à son tour la formule. Cette double inclusion ne semble pas fonctionner ici.

6. Webographie

Le projet J_LA_TE_XMath est accessible à l'adresse <http://forge.scilab.org/index.php/p/jlatexmath/>.

Le site du projet **DocBook** est <http://docbook.sourceforge.net/>.

Une référence **indispensable** sur **DocBook** et l'adaptation des feuilles de styles est <http://www.sagehill.net/docbookxsl/>. On y trouve une documentation claire et complète dispensée par un professionnel, par ailleurs développeur des feuilles de styles.

EMACS est l'éditeur de texte utilisé pour composer ce document. Le site est <http://www.gnu.org/software/emacs/>. Le mode nXML de cet éditeur est un paquetage complémentaire disponible à l'adresse <http://www.thaiopensource.com/nxml-mode/>.

7. GNU Free Documentation License

See <http://www.gnu.org/licenses/gpl-3.0.html>.