View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by CERN Document Server

EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN–AS/98–001
CERN–IT/98/2
30 mars 1998

Afficher les documents scientifiques sur le Web

Michel Goossens[a)] et Jean-Yves Le Meur[b)]
CERN, CH-1211 Genève 23, Suisse.

Résume

Le CERN traite chaque jour un très grand nombre de documents de recherche, principalement balisés en LaTeX et provenant de divers serveurs Internet. Notre but est de les afficher sur le Web via le *Catalogue des Preprints* de la Bibliothèque du CERN sous plusieurs formes (PostScript, PDF, GIF). Nous passons en revue la procédure de conversion et décrivons nos essais de production massive pour générer de l'HTML directement à partir des sources TeX. Nous finissons par une discussion de quelques développements récents dans le cadre de XML (et MML) qui amélioreront le support pour les formules mathématiques dans les programmes de visualisation.

Abstract

CERN daily handles a large number of research documents, mostly marked up in LaTeX and coming from many Internet servers. Our aim is to make them easily locatable on the Web with the help of the CERN Library's *Preprint Catalogue* in several formats (PostScript, PDF, GIF). We review the conversion procedures and give some details on some massive production trial runs to directly generate HTML from the TeX sources. We conclude with a discussion of recent developments in the framework of the XML (and MML) efforts which should ease the support of mathematics formulae in Web browsers.

Présenté à la dixième conférence TeX européenne
à Saint-Malo du 29 mars – 1 avril 1998
Publié dans les actes : Cahiers GUTenberg *28–29, pages 181–196.*

a.) IT Division, <Michel.Goossens@cern.ch>
b.) AS Division, <Jean-Yves.Le.Meur@cern.ch>

1 Le catalogue des preprints du CERN

1.1 Introduction

Le *catalogue des preprints* est un des plus volumineux catalogues de la bibliothèque du CERN. Cette « littérature grise », constituée d'articles soumis pour publication, utilise le même système informatique pour cataloguer que celui des livres, périodiques, vidéos, coupures de presse, etc. Les documents peuvent provenir directement du CERN ou de n'importe quel autre institut dans le monde et ils doivent être rapidement, durablement et facilement accessibles sur le World Wide Web.

L'acquisition de ces documents est devenue de plus en plus électronique ces dernières années et l'objectif est d'approcher une acquisition 100% automatisée.

Dans cette optique, nous considérons un document comme constitué des éléments suivants :

- une information bibliographique (aussi appelée « méta-données ») ;
- un texte en papier (le document complet et imprimé) ;
- un texte électronique (le format est variable) ;
- les figures électroniques (non incluses dans le texte principal).

Chacun de ces éléments suit un traitement particulier.

1.2 Quantité et flux

Environ deux cent mille documents de recherche sont stockés et plus de trente mille sont en format électronique. Chaque année, douze mille nouveaux documents sont ajoutés ce qui représente mille quatre cents pages par jour, ou encore une page par minute. Parmi ces documents, une grande majorité sont en TEX ou LATEX.

Trois flux réguliers de documents alimentent la collection des preprints :

- ceux en provenance du CERN ;
- ceux en provenance de l'archivage électronique de Los Alamos ;
- ceux reçus sur support papier par courrier du monde entier.

Le traitement est différent selon l'origine, mais l'objectif est toujours le même : permettre de rechercher l'information bibliographique, visualiser le texte et l'imprimer depuis toute plate-forme (X-Window, PC et Mac).

1.3 Documents du CERN

Le traitement le plus complexe est celui appliqué aux preprints créés au CERN même ! Ceci est dû d'abord à la diversité des formats utilisés au CERN (les principaux formats sont TEX, Word et FrameMaker), et à l'absence de toute politique concernant l'usage d'éditeurs standards. Puis, en tant que créateur du document et quelquefois son unique distributeur, nous avons la complète responsabilité de sa bonne diffusion.

L'auteur ou le responsable envoie le document source et les méta-données à l'aide d'un formulaire sur le Web au serveur des preprints. Les méta-données sont formatées pour être ajoutées à la base de données (ALICE) où certains champs sont indexés (titre, auteurs, numéro du rapport, date, résumé, etc.). Puis le document source est converti en PostScript et en PDF qui sont ensuite rendus disponibles par leur enregistrement dans la base de données. Des contrôles de qualité (au niveau du format, non du contenu) sont réalisés par les services de la bibliothèque (voir partie gauche de la figure 1).

1.4 Documents de l'archive électronique de Los Alamos

Il existe en physique des hautes énergies une archive centralisée à Los Alamos pour les documents de recherche, qui y sont soumis directement par les auteurs. Ces documents sont récupérés au CERN dans leur format d'origine (souvent sous forme d'archive compactée `tar.gz`) pour être transformés dans un format lisible par tous.

Deux procédures sont exécutées en parallèle : l'une traite via un abonnement courrier électronique (E-mail Subscription) les méta-données alors que l'autre récupère et convertit les sources (Electronic Fulltext) des documents (voir partie centrale de la figure 1).

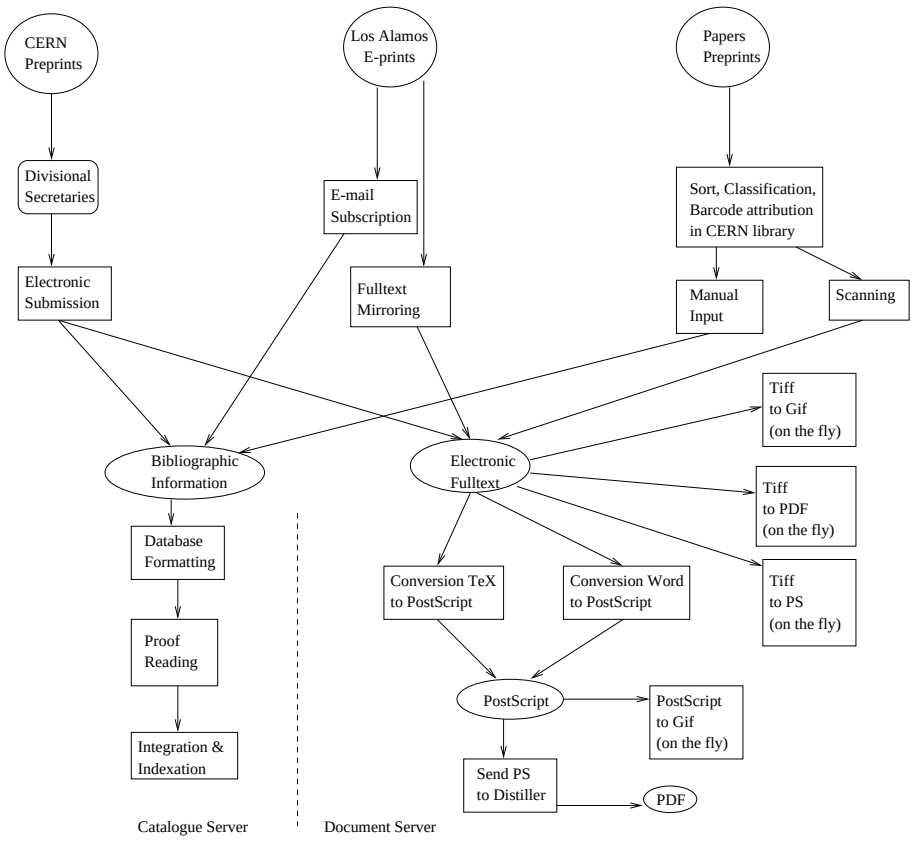

FIG. 1 – *Flux des données pour l'archivage de preprints au CERN*

1.5 Documents papier

Une quantité de plus en plus petite de documents sont envoyés par la poste à la bibliothèque centrale du CERN. Lorsqu'aucune version électronique n'est disponible et que le document est sélectionné pour diffusion, la procédure décrite dans la partie droite de la figure 1 est appliquée.

Après sélection, les documents sont digitalisés pour être conservés en format TIFF (compression G4). L'information bibliographique est entrée manuellement dans le catalogue. Plusieurs logiciels de reconnaissance optique de caractères (OCR) ont été testés afin de récupérer le résumé mais malgré des résultats assez positifs, la qualité du texte obtenu ne permet pas de l'injecter automatiquement dans la base de données. Ceci est dû en majeure partie à la présence fréquente de symboles mathématiques dans le texte, ce qui rend la reconnaissance des caractères problématique, voire impossible.

1.6 Les conversions de formats

1.6.1 TEX à PostScript

Cette procédure est bien connue. Son automatisation est réussie dans 98% des cas, l'objectif étant la création d'un document PostScript correct sans aucune intervention humaine.

La première étape identifie la version du TEX utilisée afin de lancer la conversion appropriée (`tex` ou `latex`). La conversion produit un fichier DVI.

La seconde étape transforme le format DVI en PostScript à l'aide de l'utilitaire `dvips`. À ce niveau, deux fichiers PostScript sont créés : le premier est compacté (gzip) et rendu disponible sur le serveur, le second est créé en incluant les polices en format *Type 1* pour être transféré vers le serveur de conversion de PostScript en PDF. Cette inclusion des polices est nécessaire à la création de documents PDF de bonne qualité.

Les échecs encore rencontrés concernant cette conversion sont liés à l'usage de macros TEX créées par les auteurs et que nous ne possédons pas.

Les conversions directes de TEX vers PDF (`pdfTeX`) ou de DVI vers PDF (`dvipdf`) ne sont

pas encore dans un état d'avancement suffisant pour être appliquées directement et raccourcir ainsi la procédure. Aussi créons-nous le PDF depuis le PostScript.

1.6.2 De PostScript à PDF

La *distillation* du PostScript en PDF se fait chaque douze heures en envoyant les fichiers PostScript sur un répertoire monitoré. La création des fichiers PDF est ainsi complètement automatique.

Un échec dans cette conversion signifie généralement une inconsistance dans le fichier PS, ce qui nous permet en même temps de détecter des fichiers PostScript corrompus.

1.6.3 De `Word` à PostScript/PDF

C'est au niveau de cette conversion que l'automatisation est pour l'instant la moins fiable. Une tentative a été faite de transférer vers un serveur (Mac) tout document `Word` nous parvenant et d'y simuler une impression pour générer le PostScript. Cela a été un échec pour deux raisons :

- les versions de `Word` encore en usage, sur Mac ou PC ne sont pas pleinement compatibles ;
- les logiciels de dessin utilisés par les auteurs, dont les images sont ensuite « copiées/collées » dans le document `Word`, sont souvent inconnus ou inaccessibles sur le serveur.

Nous testons actuellement des logiciels qui permettent la conversion en traitement par lots de `Word` vers PostScript mais rien de vraiment satisfaisant n'est encore apparu. Le passage par le RTF (*Rich Text Format*) permet l'échange de l'information textuelle entre plusieurs plates-formes mais ne garantit pas l'accès aux figures.

Les documents Word représentent heureusement actuellement une minorité des documents transmis et un support manuel peut être apporté à cette tâche.

1.6.4 Les conversions à la volée

Afin de minimiser le stockage et la maintenance de documents en différents formats, la conversion à la volée permet d'offrir à partir d'un format unique une variété d'autres formats.

Cette technique est utilisée pour délivrer les formats suivants :

- TIFF vers PostScript : la conversion produit du PostScript niveau 2. Chaque document scanné est ainsi disponible en PostScript.
- TIFF vers PDF : le PDF créé est une image (plan de bits). Toutes les options que PDF permet (recherche de texte, copier/coller, etc) ne sont donc pas disponibles suite à cette conversion mais la visualisation et l'impression peuvent se faire via *Acrobat Reader* d'Adobe.
- TIFF et PostScript vers GIF : ces conversions visent à rendre possible la visualisation des documents scannés directement à l'intérieur du visualisateur (sans extension — *plug-in*), page par page.

2 Le chemin direct de TEX vers le Web

Notre objectif est de permettre la lecture des preprints du catalogue CERN en HTML. La visualisation d'un document écrit en TEX sur un visualisateur Web peut se faire selon plusieurs procédés (liste non exhaustive !) :

- une conversion complète en HTML du document en traitement par lots (p. ex. `LaTeX2HTML` et `TeX4ht`) ;
- une conversion complète en HTML du document à la volée (p. ex. `tth`) ;
- une conversion partielle en traitement par lots, combinée à une application Java (p. ex. `idvi`) ;
- une lecture directe du source LATEX par une application Java (p. ex. `WebEQ`) ;
- une lecture directe du source LATEX par une application *ad-hoc* (p. ex. l'utilitaire `techexplorer`).

Pour donner une idée de la qualité de traduction nous reproduisons dans ce qui suit pour chaque programme décrit le résultat après traitement du même fragment d'un document type représentatif des formules de physique que nous avons à traiter habituellement.

2.1 Conversion en traitement par lots : LaTeX2HTML

2.1.1 Description

`LaTeX2HTML` est un programme écrit en perl, qui fait appel à des utilitaires de la bibliothèque `ghostscript` pour la conversion des images. En général plusieurs fichiers HTML sont créés, et une table des matières permet une navigation aisée. Chaque image et certaines constructions mathématiques complexes sont converties en images GIF (ou PNG). La dernière version officielle est disponible à l'URL `http://www-dsed.llnl.gov/files/programs/unix/latex2html/`.

2.1.2 Avantages

La qualité. La qualité de l'HTML produit par `LaTeX2HTML` est indéniable (voir les figures 2 et 3). La mise en page et le système de navigation sont performants. Des liens sont créés vers les références. Les équations et les figures, converties en GIF, sont très lisibles. Il y a un large choix d'options pour contrôler la traduction. Pour les formules on peut les transformer « globalement » (en une partie, c'est le cas de la figure 2) ou rechercher des parties de formules qui peuvent être exprimées en HTML directement (c'est le cas de la figure 3). Il faut souligner que la *consistance* de la notation est très importante et que `LaTeX2HTML` y attache une importance extrême surtout dans le deuxième cas, où la représentation des variables à l'intérieur des images GIF et en HTML doit être compatible (p. ex. les variables doivent être en italique dans les formules et le texte). Nous remarquons toutefois quelques imprécisions dans le placement des différents sous-éléments constituant une formule, un comportement que nous retrouverons dans les approches `TeX4ht` et `tth`, qui utilisent des stratégies similaires.

Fiabilité et lisibilité. `LaTeX2HTML` a été largement utilisé avec succès sur des fichiers TEX ou LATEX. Le résultat est très fiable. `LaTeX2HTML` génère des fichiers qui contiennent un HTML « normalisé », ce qui les rend lisibles par tous les programmes de visualisation. En particulier, même avec un visualisateur qui ne représente que la partie textuelle (p. ex. Lynx) on obtiendra un résultat plus qu'acceptable.

2.1.3 Inconvénients

Le temps. `LaTeX2HTML` a été intégré en test dans le cycle de production-distribution des preprints au CERN.

En moyenne, la conversion d'un article d'une dizaine de pages peut prendre plusieurs minutes temps réel (sur un poste de travail performant), ce qui présente quelques problèmes vu le nombre important de documents à traiter. À plus forte raison, cette lenteur d'exécution — principalement due à la conversion des images en GIF — rend également impossible son utilisation à la volée.

L'espace disque. L'obligation de collecter et de stocker les nombreux fichiers HTML ainsi que les images GIF est une lourde contrainte. La procédure requiert elle aussi un doublement voire plus, de l'espace de stockage pour un document donné.

La vitesse de chargement. Le fait que les équations soient traduites en GIF ralentit d'une façon significative la vitesse de chargement des pages riches en formules. C'est le prix à payer pour garantir l'intégrité des formules converties.

L'installation. Le script `LaTeX2HTML`, écrit en `perl`, fait appel à de nombreux utilitaires publics dont l'installation peut paraître fastidieuse, voire impossible, à un non-spécialiste (surtout sur PC et Mac).

2.2 Conversion en traitement par lots *bis* : TeX4ht

2.2.1 Description

`TeX4ht` remplace les commandes TEX par des séquences HTML en utilisant principalement une ou plusieurs extensions LATEX. Il ne nécessite donc aucun système externe (comme `perl`

$$\phi_v\left(\lambda_v,\kappa,\beta^2\right) = \frac{1}{2\pi i}\int_{c-i\infty}^{c+i\infty}\phi(s)\,e^{\lambda s}ds \qquad c \geq 0$$

$$\phi(s) = \exp\left[\kappa(1+\beta^2\gamma)\right]\ \exp\left[\psi(s)\right],$$

$$\psi(s) = s\ln\kappa + (s+\beta^2\kappa)\left[\ln(s/\kappa) + E_1(s/\kappa)\right] - \kappa e^{-s/\kappa},$$

and

$$E_1(z) = \int_z^\infty t^{-1}e^{-t}dt \qquad \text{(the exponential integral)}$$

$$\lambda_v = \kappa\left[\frac{\epsilon-\bar{\epsilon}}{\xi} - \gamma' - \beta^2\right]$$

The Vavilov parameters are simply related to the Landau parameter by $\lambda_L = \lambda_v/\kappa - \ln\kappa$. It can be shown that as $\kappa \to 0$, the distribution of the variable λ_L approaches that of Landau. For $\kappa \leq 0.01$ the two

FIG. 2 – *Fragment généré par* `LaTeX2HTML` *(options par défaut)*

$$\phi_v\left(\lambda_v,\kappa,\beta^2\right) = \frac{1}{2\pi i}\int_{c-i\infty}^{c+i\infty}\phi(s)\,e^{\lambda s}dsc \geq 0$$

$$\phi(s) = \exp\left[\kappa(1+\beta^2\gamma)\right]\exp\left[\psi(s)\right],$$

$$\psi(s) = s\ln\kappa + (s+\beta^2\kappa)\left[\ln(s/\kappa) + E_1(s/\kappa)\right] - \kappa e^{-s/\kappa},$$

and

$$E_1(z) = \int_z^\infty t^{-1}e^{-t}dt \qquad \text{(the exponential integral)}$$

$$\lambda_v = \kappa\left[\frac{\epsilon-\bar{\epsilon}}{\xi} - \gamma - \beta^2\right]$$

The Vavilov parameters are simply related to the Landau parameter by $\lambda_L = \lambda_v/\kappa - \ln\kappa$. It can be shown that as $\kappa \to 0$, the distribution of the variable λ_L approaches that of Landau. For $\kappa \leq 0.01$ the two

FIG. 3 – *Fragment généré par* `LaTeX2HTML` *(formules fractionnées)*

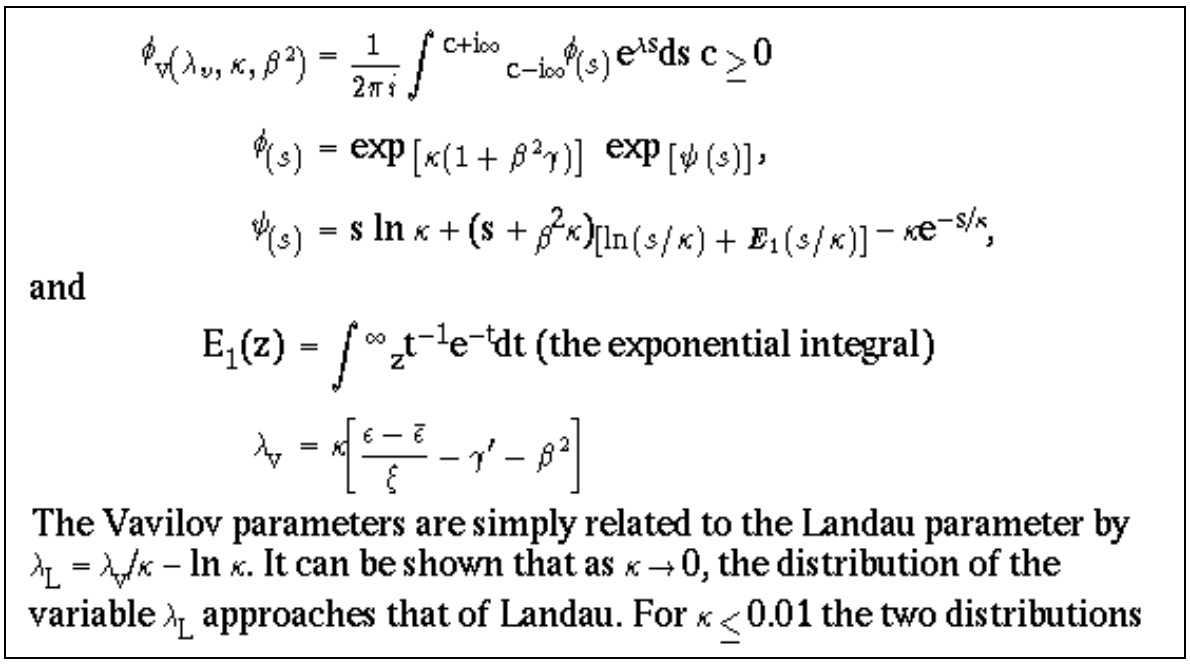

FIG. 4 – *Fragment généré par* `TeX4ht`

pour `LaTeX2HTML`), mais il utilise quand-même des utilitaires de la bibliothèque `ghostscript` pour la conversion des images et des formules mathématiques contenant des symboles spéciaux. En général il faut modifier le document source légèrement en y incluant quelques directives supplémentaires. La page de référence est à l'URL `http://www.cis.ohio-state.edu/~gurari/TeX4ht/mn.html`. La figure 4 montre notre document type traité avec `TeX4ht`.

2.2.2 Avantages et inconvénients

Les avantages et inconvénients sont essentiellement ceux de `LaTeX2HTML`. `TeX4ht` est pratiquement uniquement basé sur LaTeX, en fait appel à des extensions (en particulier `TeX4ht.sty`) et des fichiers pour les polices. L'installation s'avère donc un peu plus aisée, surtout sur les machines qui ne tournent pas sous Unix. Pour les formules, `TeX4ht` utilise des images GIF (générées avec `dvips` et `ghostscript`). Même si, comparé à `LaTeX2HTML`, la qualité (notez p. ex. le placement des limites des intégrales à la figure 4) du code généré n'est pas toujours optimale, cet outil présente néanmoins une alternative intéressante.

2.3 Conversion à la volée : `tth`

2.3.1 Description

L'application `tth` est écrite en C et n'utilise aucune application extérieure pour convertir TeX en HTML. Les formules sont traduites en HTML au lieu d'être converties en images. Une description complète est disponible à l'URL `http://venus.pfc.mit.edu/tth/tth.html`. Le fait le plus marquant est que `tth` utilise les polices `Symbol` disponibles sur un système X-Window, Mac ou PC. Par exemple, avec X-window, il suffit d'inclure la ligne

```
Netscape*documentFonts.charset*adobe-fontspecific: iso-8859-1
```

dans le fichier `.Xdefaults` pour utiliser les caractères en question avec Netscape. Par défaut, les images ne sont pas incluses dans le document généré mais sont disponibles comme des liens *externes* (il est toutefois possible de demander l'inclusion de ces images comme fichiers GIF, mais ceci complique le traitement). Le fragment du document type généré avec `tth` est montré en figure 5.

2.3.2 Avantages

La vitesse. Le programme est d'une rapidité étonnante, à tel point que le stockage peut se faire dans un répertoire groupé (`.tar`) et compressé (`.gz`) et que la conversion (`gunzip`, `tar xvf` puis

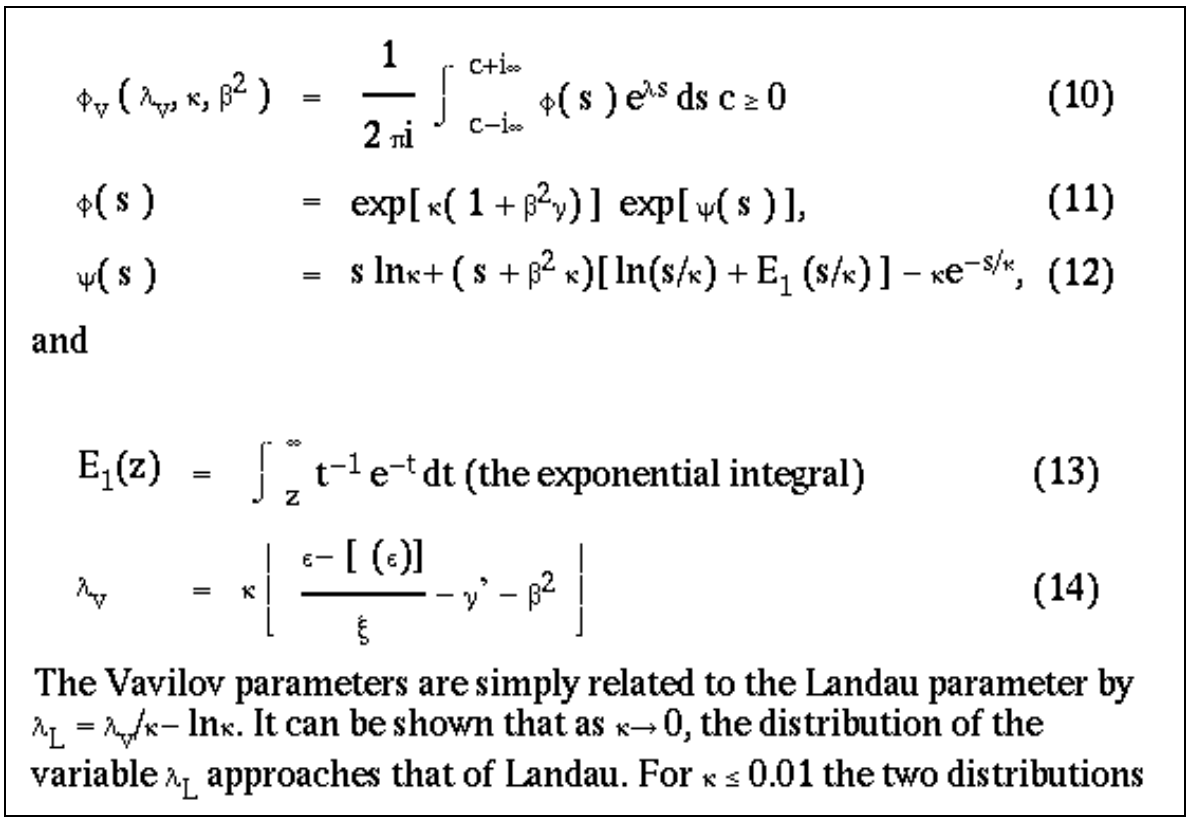

FIG. 5 – *Fragment généré par* `tth`

`tth`) peut se faire à la volée sans délai significatif.

C'est un avantage considérable pour un serveur qui stocke un très grand nombre de documents.

La maintenance. La conversion à la volée permet, dans le cas où les documents sont amenés à être modifiés, de réduire à néant la maintenance de la conversion. Comparé à une procédure en traitement par lots qui demande à être relancée quand un document est changé, le gain de temps est considérable.

2.3.3 Inconvénient majeur : la qualité

Si certains documents apparaissent proprement, il reste fréquent que `tth` échoue dans la transformation de formules complexes (les commandes pour les caractères qui ne peuvent être représentés sont copiées textuellement dans la sortie HTML). Une autre faiblesse est l'absence de références croisées, ce qui est dû au fait qu'un seul passage est effectué sur le document source.

Un point supplémentaire à noter est que pour visualiser un document généré avec `tth`, le lecteur doit adapter l'environnement de son programme de visualisation (voir ci-dessus). Même si c'est une opération simple, un lecteur non averti (ou trop impatient) sera dans l'impossibilité de lire le document correctement.

2.4 Conversion partielle : `idvi`

2.4.1 Description

Le programme `idvi` utilise un utilitaire pour préparer un document LaTeX à sa visualisation sur le Web. Une fois cette étape franchie, des applets Java permettent de lire le document exactement comme il a été formaté en TeX (mêmes pages, même présentation). Une fenêtre-menu, en plus de l'écran principal, permet de naviguer de page en page.

Une description complète de ce traducteur est disponible à l'adresse `http://www.geom.umn.edu/java/idvi/index.html`.

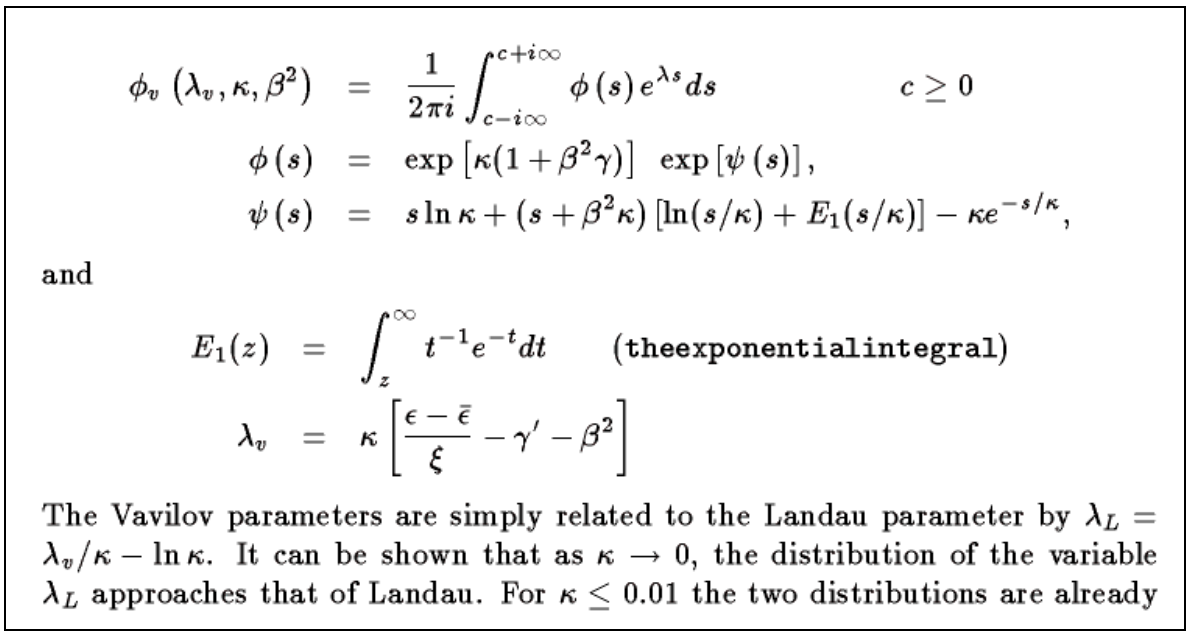

FIG. 6 – *Fragment généré par* `idvi`

2.4.2 Avantages

Qualité et fidélité. Comme tout est basé sur le fichier DVI et traité avec les polices TEX, la lisibilité du document est parfaite et les formules sont traduites de manière complètement fiable (voir figure 6). En fait `idvi` peut être considéré comme un visualisateur Java de DVI. Chaque page HTML est préparée séparément, et elle inclut l'appel à une classe Java.

Le résultat est complètement fidèle à la version écrite en TEX par l'auteur. Le résultat s'apparente au résultat obtenu en visualisant le document en PDF avec l'utilitaire d'Adobe *Acrobat Reader*.

2.4.3 Inconvénients

Vitesse de chargement et problèmes d'impression. Comme souvent avec les applications Java, le téléchargement des classes est lent. En plus, l'interprétation du code Java nécessite des ressources en CPU non négligeables sur la machine cliente où tourne le programme de visualisation.

Un document présenté avec `idvi` n'est pas imprimable. Une version DVI (ou PDF) doit donc être gardée en parallèle avec les fichiers HTML propres à `idvi`.

Portabilité du code Java. En plus de sa lenteur, l'existence de différentes versions de Java, qui ne marchent pas toujours convenablement avec les applets générées par `idvi`, font que `idvi` ne pourra être conseillé pour la consultation massive des archives de documents. Il sera probablement supplanté bientôt par `WebEQ`, que nous décrirons ci-dessous.

3 En attendant MML

Dans un futur pas trop lointain (un ou deux ans?) nous nous attendons à ce qu'un support pour MML (*Mathematical Markup Language*, voir l'URL `http://www.w3.org/TR/WD-math`) soit intégré dans les programmes de visualisation. Ainsi les formules mathématiques seront traitées au même niveau que les parties textuelles d'un document (recherches, copier/coller, réutilisation, base de données, balises normalisées). En attendant deux approches permettent de visualiser un document TEX, en le transformant en `WebTeX`, le langage de balisage de `WebEQ`, ou en traitant directement le fichier source LATEX avec `techexplorer`.

3.1 Via l'extension `techexplorer`

`techexplorer` est conçu comme une extension (*plug-in*) pour les visualisateurs Netscape et Microsoft Explorer. Actuellement des versions pour Windows 95/NT, Sun et RS-6000 sont

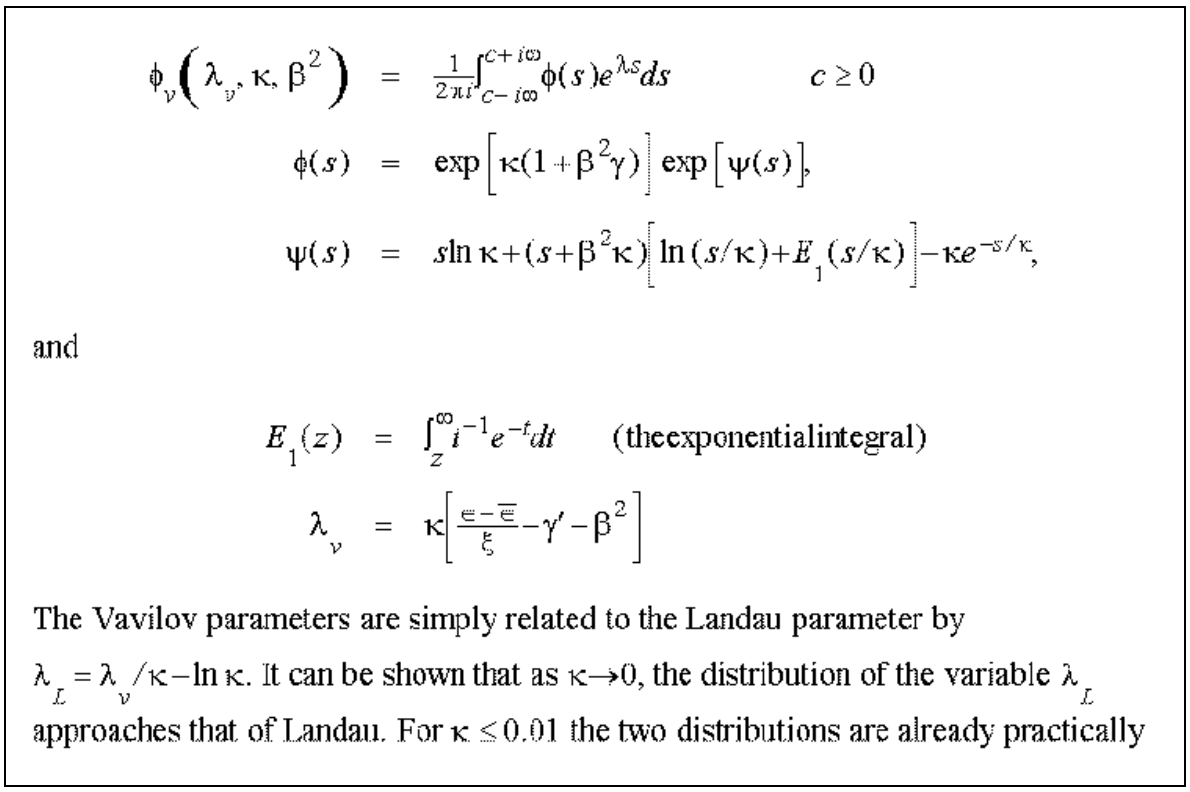

FIG. 7 – *Fragment visualisé par* `techexplorer`

disponibles et la documentation peut être consultée à URL `http://www.ics.raleigh.ibm.com/ics/techexp.htm`.

`techexplorer` peut lire directement un fichier source LaTeX, même s'il ne reconnaît pas (encore) la totalité des commandes LaTeX. En fait, `techexplorer` représentera « tel quel » (en rouge) les séquences de commande LaTeX qu'il ne sait pas traiter directement. En plus de ces commandes LaTeX, `techexplorer` offre toute une panoplie d'extensions hypertext et multimédia, qui en font un outil performant pour préparer des documents pour le Web. La figure 7 montre le même fragment du document utilisé auparavant. Ici aucune préparation n'est nécessaire, le document source LaTeX est présenté au visualisateur directement. Un support direct pour MML a été promis pour les prochains mois.

3.2 Via Java : `WebEQ`

`WebEQ` est un système de visualisation d'équations pour le Web utilisant Java pour rendre les formules. `WebEQ` a son propre éditeur pour les équations, dont l'interface de la dernière version (2.2) pour Windows NT est montrée dans la figure 8. La fenêtre en haut à gauche montre une partie des formules de notre document, à droite le menu permettant le copier-coller des symboles spéciaux, et finalement à gauche en bas, le début de la sortie MML correspondant à la fenêtre de sortie. Cet éditeur permet de composer directement les formules à l'écran ou de lire un ficher source en `WebTeX`, un langage qui combine HTML pour le texte et un balisage inspiré de TeX, mais d'une fonctionalité réduite pour les mathématiques (les formules sont saisies *à la TeX* entre `$...$` ou `\[...\]` et un programme spécifique (le *Sizer*) les traduit en applets Java permettant la visualisation avec `WebEQ`.

C'est en appliquant cette procédure que nous avons obtenu le fragment de notre document type montré à la figure 9. Ainsi l'éditeur de `WebEQ` est un outil idéal pour s'initier à MML, comme il permet de traduire des formules (pas trop compliquées) de LaTeX en MML. Plus d'information sur `WebEQ` se trouve à l'URL `http://www.geom.umn.edu/software/WebEQ/`.

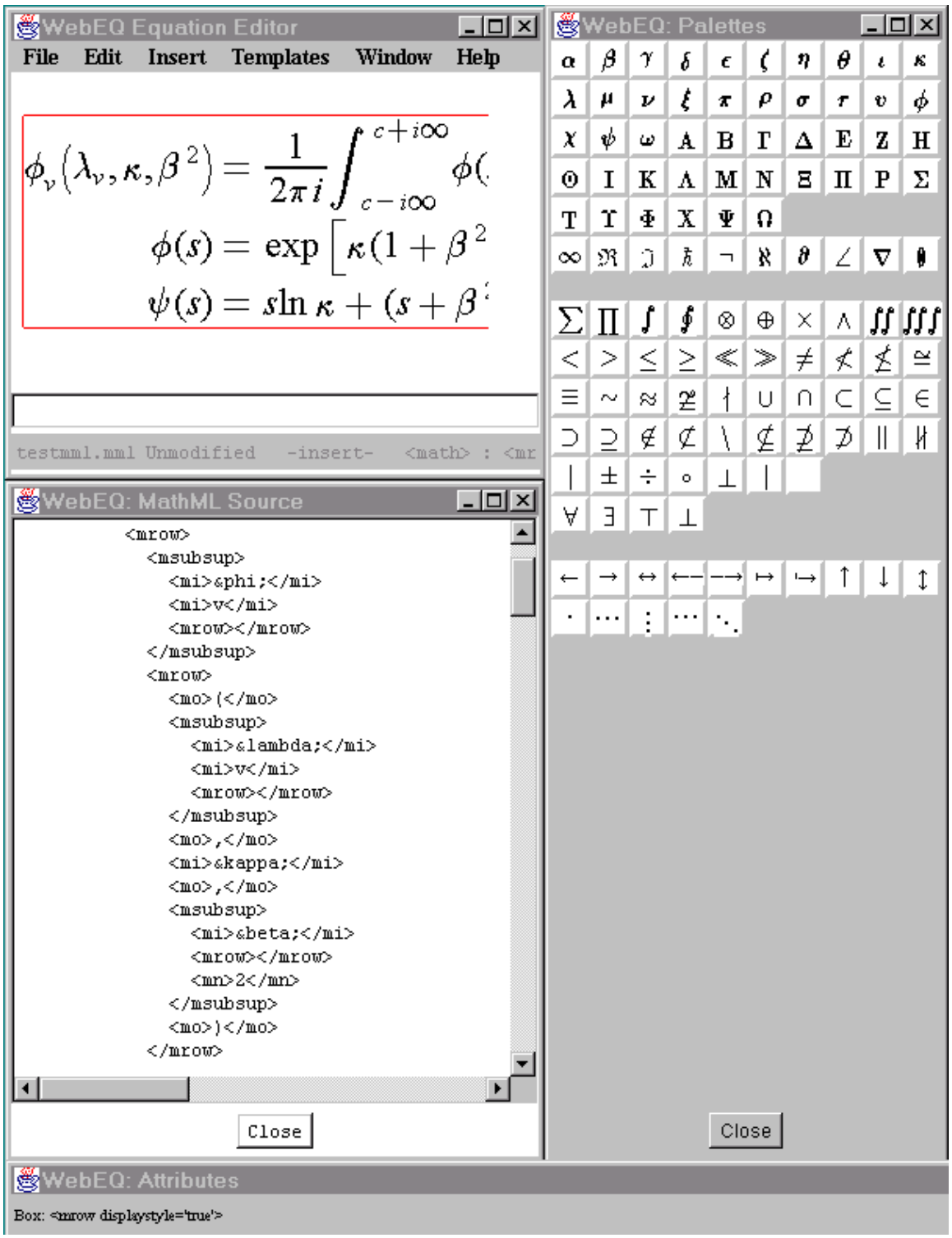

FIG. 8 – *L'éditeur équations de* `WebEQ`

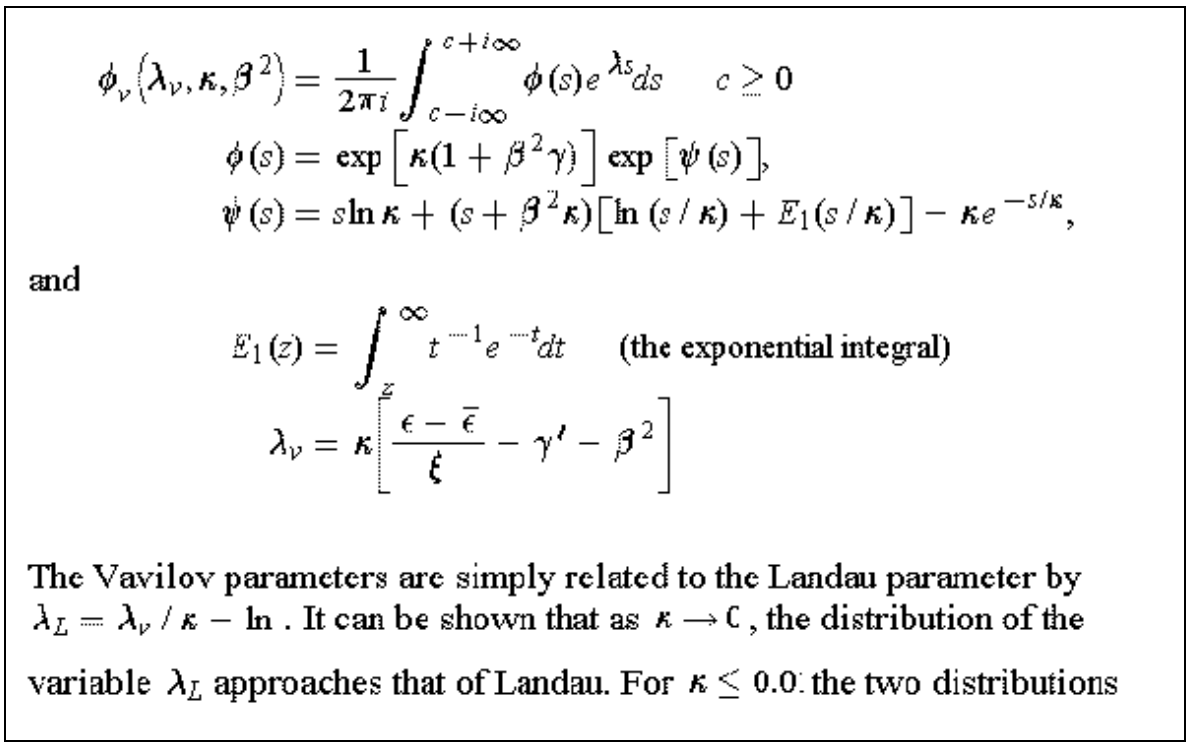

FIG. 9 – *Fragment généré par* `WebEQ`

4 Conclusions

Préparer des documents scientifiques pour les visualier sur le Web n'est pas toujours trivial, surtout quand l'information contient des parties mathématiques. Les problèmes sont maintenant bien connus (voir p. ex. à l'URL `http://forum.swarthmore.edu/typesetting/index.html`) et les deux seules solutions complètes sont soit une sortie PDF (ou PostScript), soit une conversion de toutes les formules en images. C'est la philosophie (plus ou moins) adoptée par presque tous les programmes qui transforment un fichier source LaTeX en ces formats. Toutefois dans un cadre de production et de distribution massive de documents, la solution 100% HTML n'est pas encore fiable.

Le développement et la mise à disposition générale de XML et MML dans les deux prochaines années assureront que les parties « formules » (mathématiques, chimie, etc.) pourront être traitées de manière plus satisfaisante par les programmes de visualisation (et les bases de données pour l'extraction de l'information). En attendant, les approches hybrides, comme celles évoquées dans cet article, permettront au plus grand nombre de profiter d'une distribution quasi-instantanée de l'information sur l'Internet.