

Presenting Mathematics and Languages in Web-pages, using L^AT_EX2HTML

Ross Moore

Mathematics Department

Macquarie University

Sydney, Australia 2109

ross@mpce.mq.edu.au

<http://www-math.mpce.mq.edu.au/~ross/>

Introduction

L^AT_EX2HTML is a very flexible tool for creating Web pages to display the information contained in a manuscript prepared using L^AT_EX. As of July 1998, the current version is L^AT_EX2HTML v98.2. It runs under Unix, Linux, Windows NT, Windows'95, OS/2 and DOS. The latest released version, with online manual for browsing, can be obtained from its distribution site¹, in the USA or the European mirror². On CTAN, look under `support/latex2html`. There is a developers repository³ for minor updates and (α - or β -) development versions.

In the following sections we first discuss some general considerations for Web pages using HTML, including some pragmatic tips for authors wishing to use the L^AT_EX2HTML translator. This is followed by a study of the different ways that are available for the presentation of mathematics using L^AT_EX2HTML, discussing the available options and when a particular approach may be most appropriate. Further examples are presented on how to use L^AT_EX2HTML to produce multi-lingual documents.

General considerations: Why HTML?

For distribution of text-like data on the Internet the HTML formats, in their various versions, are very efficient and widely supported in Web browser software on all computing platforms. Thus a converter that produces documents using HTML can guarantee that the information to be presented is accessible to the widest possible audience. Furthermore, there is no requirement for 'plug-in' modules or other special software, beyond what is normally available with a Web browser.

Since an `.html` file contains just editable text, it is easily modified in any editor. This property alone adds a significant level of flexibility to any transla-

tion tool. If the result of the automatic translation is not quite what is desired, it is a simple matter to find the place where a correction is necessary and do it 'by hand'.

Even if the translation is flawless, at some time in the future there may be a change desired in the information being presented. For example a name or address may change, or a different graphic image may be desired, or a hyperlink to some external site may become invalid so needing to be replaced by another. Such minor alterations and updates can be done *without the need to reprocess the whole document* from the original L^AT_EX source (which may no longer even be available).

Another flexible aspect of HTML is that the reader has control over the browser window's characteristics. This includes size, style and colour of the text-font in which most of the information is to be presented, as well as the location and shape of the viewing window. The reader can customise these properties to suit personal requirements and preferences. This is a feature not available with other data formats, such as `.dvi`, `.pdf` or PostScript.

Mathematics with L^AT_EX2HTML

Figure 1 shows how pieces of mathematics may be presented, using L^AT_EX2HTML's default settings for the versions released during 1997 and 1998. This is a 'screen-shot' of a portion of a Web page generated using L^AT_EX2HTML. Fuzziness in the image is due to the lower resolution for on-screen display than is typically used with a printed version. Furthermore "anti-aliasing" is used with the font characters, to avoid a jagged appearance.

The L^AT_EX code for this example is given at the end of this article. It displays many common features of typeset mathematics:

- Greek letters and calligraphic (script) symbols;
- superscripts, subscripts, fractions and derivatives;
- large operators, such as \int and \sum with limits;

¹ <http://www-dsed.llnl.gov/files/programs/unix/latex2html/>

² <ftp://ftp.rzg.mpg.de/pub/software/latex2html/>

³ <http://cdc-server.cdc.informatik.tu-darmstadt.de/latex2html/>



Math examples

$$\phi(\lambda) = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} \exp(u \ln u + \lambda u) du \quad \text{for } c \geq 0 \quad (1)$$

$$\lambda = \frac{\epsilon - \bar{\epsilon}}{\xi} - \gamma' - \beta^2 - \ln \frac{\xi}{E_{\max}} \quad (2)$$

$$\gamma = 0.577215 \dots \quad (\text{Euler's constant}) \quad (3)$$

$$\gamma' = 0.422784 \dots = 1 - \gamma \quad (4)$$

$$\bar{\epsilon} = \text{average energy loss} \quad (5)$$

$$\epsilon = \text{actual energy loss} \quad (6)$$

Since (2) or (3) should hold for arbitrary $\delta \mathbf{c}$ -vectors, it is clear that $\mathcal{N}(A) = \mathcal{R}(B)$ and that when $y = \mathcal{B}(x)$ then one has ...

$$V \pi^{sr} = \left\langle \sum_i M_i \mathbf{V}_i \mathbf{V}_i + \sum_i \sum_{j>i} \mathbf{R}_{ij} \mathbf{F}_{ij} \right\rangle \quad (7)$$

$$= \left\langle \sum_i M_i \mathbf{V}_i \mathbf{V}_i + \sum_i \sum_{j>i} \sum_{\alpha} \sum_{\beta} \mathbf{r}_{i\alpha j\beta} \mathbf{f}_{i\alpha j\beta} - \sum_i \sum_{\alpha} \mathbf{P}_{i\alpha} \mathbf{f}_{i\alpha} \right\rangle$$

$$B_{ij}^{\alpha} = (B_{ij}^{\alpha})_0 + (B_{ij}^{\alpha})_a \quad (8a)$$

$$(B_{ij}^{\alpha})_0 = \frac{1}{2} \left(\frac{\partial N_i^{\alpha}}{\partial X_j} + \frac{\partial N_j^{\alpha}}{\partial X_i} \right) \quad (8b)$$

$$(B_{ij}^{\alpha})_a = H_{ij}^{\alpha\beta} a^{\beta} \quad (8c)$$

Figure 1: Some mathematics in a Web document produced by L^AT_EX₂HTML, using the default settings.

- aligned arrays of equations (in particular the `subequations` environment from the `amsmath` package);
- extended brackets and parentheses.

Notice how most of the mathematics looks just like it has been typeset in T_EX, because that is precisely what has happened. These expressions are actually images, in GIF format.⁴ This is achieved by typesetting each mathematical expression on a single page, processing the `.dvi`-file using `dvips`, then rendering

⁴ Alternatively images can be generated using the PNG graphics format.

the resulting PostScript⁵ files using *Ghostscript*.⁶ For a better quality on-screen appearance in these low-resolution images, the ‘anti-aliasing’ technique is employed to soften the edges of otherwise ‘blocky’ font characters. For readability on-screen, images of mathematical expressions are normally made to correspond to a 14 pt font-size. Like most choices in L^AT_EX₂HTML, this can be altered.⁷

⁵ PostScript is a registered trademark of Adobe Systems Inc.

⁶ *Ghostscript* is a product of Aladdin Enterprises, Menlo Park, CA. Version 4.02 or later is required for ‘anti-aliasing’ effects.

⁷ ... by adjusting the value of the `$MATH_SCALE_FACTOR` configuration variable.

‘Simple math’. Some of the mathematical expressions in figure 1 do *not* use an image; e.g., the inline expression $y = B(x)$ and most of the ‘=’ signs in the first alignment. For these the whole expression can be represented using ordinary font characters, so this is what is done—with names set in italics, of course.

If any special symbol, indeed any macro (apart from those listed below), occurs within a mathematical expression then an image is made of the whole expression. To $\text{\LaTeX}2\text{HTML}$ users this is known as the “simple math” strategy. Superscripts, subscripts and some simple type-face macros are handled appropriately. Allowable macros include `\mathbf`, `\mathrm`, `\mathtt`, `\mathit` and `\boldsymbol`, as well as the recent `\mb` addition to \LaTeX . Furthermore `\textbf`, `\textrm`, `\texttt`, `\textit` are allowable, but not recommended.

Although the appearance of expressions presented using ‘simple math’ are generally not as attractive as with an image, the benefit is that less information needs to be transferred across the network. For example, the expression $\mathcal{N}(A) = \mathcal{R}(B)$ results in HTML code:

```
<!-- MATH:  $\mathcal{N}(A) = \mathcal{R}(B)$  -->
<IMG
  WIDTH="104" HEIGHT="31" ALIGN="MIDDLE" BORDER="0"
  SRC="img14.gif"
  ALT=" $\mathcal{N}(A) = \mathcal{R}(B)$ ">
```

Notice how the \TeX source is included as a comment. This ensures that the information is available in the `.html` file, in case the image fails to load successfully. When sufficiently short, the source is also included within the ALT attribute of the `` tag. This allows a textual representation to be shown by browsers which do not support images (e.g., *lynx*) or when image support has been deliberately disabled.

Compare this with the amount of code generated for $y = B(x)$, using ‘simple math’:

```
when <I>y</I>=<I>B</I><I>x</I> then one has
```

It is not difficult to appreciate the advantages to this ‘simple math’ approach. Furthermore, in the previous paragraph there was no mention of the actual size of the file `img14.gif` that needs to be transferred to show the image, and the extra server-connection required to request it be sent. In practice these are more significant than the extra text required within the `.html` file. Furthermore images do not rescale automatically when the font-size is changed within the browser.

It is clear that ‘simple math’ is a good strategy when a Web document contains only simple mathematical expressions, for then the overhead to request and transfer images is minimal. However when a lot

of quite complicated mathematics is to be presented, this approach is not ideal. We later discuss alternative strategies available with $\text{\LaTeX}2\text{HTML}$.

Alignment environments. Equation alignments are achieved using HTML’s `<TABLE>` tag. Such tags became available as a standard part of HTML with the version 3.2 recommendation in early 1997. Some browsers provided support earlier than this.

Each cell in the table is treated as a separate expression, for deciding whether to use “simple math” or to make an image. Compare the different size of the ‘=’ signs in the equation beginning $\gamma' = \dots$ in figure 1. The first uses the browser’s font whereas the second one is part of an image.

Notice also that equation numbers are placed in a separate column of cells within the `<TABLE>`. The `leqno` document-class option causes numbering to be put on the left-hand side, as with \LaTeX .

Overriding ‘simple math’. The default ‘simple math’ strategy can be turned-off using the `-no_math` command-line switch. That is, run $\text{\LaTeX}2\text{HTML}$ on the \LaTeX source file using the command:

```
latex2html -no_math ... myfile.tex
```

where the ‘...’ indicate the possible presence of other command-line switches. This will provide a consistent style for the mathematical expressions in all parts of the environment, as in an on-paper typeset version. This can be seen in figure 2, showing the inline mathematics portion of figure 1.

Since (8) or (8d) should hold for arbitrary $\delta\mathbf{c}$ -vectors, it is clear that $\mathcal{N}(A) = \mathcal{R}(B)$ and that when $\mathbf{y} = B(\mathbf{x})$ then one has ...

Figure 2: Inline mathematics, without using the ‘simple math’ strategy. An opaque background shows the size of images; extra space is included to allow correct alignment. (Normally image backgrounds are transparent.)

There will be more images than when ‘simple math’ is used. An appropriate situation for this strategy might be when the complete HTML document is available on the local machine or network (LAN), so that expensive file-transfers are not an issue.

Normally images are created with transparent backgrounds.⁸ In figure 2, an opaque background has been used to show the size and alignment of the images, with respect to the surrounding text. Notice

⁸ This is overridden by the `-no-transparent` command-line switch.

that when there is a ‘descender’ the image contains extra white space below the baseline. This allows the `` attribute to position the image correctly. With no descender `` is appropriate.

The extra height causes wide line-spacing in older browsers. This anomaly can be fixed for more recent browsers, by using the `.css` stylesheet[7] that `LATEX2HTML` produces automatically. One needs to set the `line-height` property to a fixed amount; e.g.,

```
P.INLINE      { line-height : 20pt }
```

Now within the `.html` pages, change the `<P>` tag to `<P CLASS="INLINE">` for paragraphs containing over-sized images. The technique was used with figure 8. Future versions of `LATEX2HTML` will handle this automatically, at least when preparing code according to HTML 4.0 specifications.

Images of aligned environments. In earlier versions of `LATEX2HTML` an image was made of whole `eqnarray` and `equation` and other environments. Before `<TABLE>` tags were recommended within the HTML 3.2 standard, this was necessary and equation numbering was included as part of the image. Now this effect can be achieved, when desired, in several different ways.

Easiest is to request that `LATEX2HTML` produce HTML code conforming to the version 2.0 standard, using the command-line option:

```
latex2html -html_version 2.0 ... myfile.tex
```

However this will disallow other constructions; e.g. forcing images also of `tabular` environments. Using also `-no_math` ensures images of all inline formulae as well.

Alternatively, images can be forced selectively by including an `\htmlimage` command within the environment. This command takes an argument which allows extra graphic effects to be specified for the image; see the User Manual[1] for the available effects:

```
\begin{eqnarray}
\htmlimage{
...
...
\end{eqnarray}
```

Finally, the `makeimage` environment creates an image of whatever `LATEX` code it contains. Both this and the `\htmlimage` command require the `html` package be loaded within the document preamble.

```
\begin{makeimage}
\begin{eqnarray}
...
...
\end{eqnarray}
\end{makeimage}
```

Image Reuse and Reduction Strategies

A document such as a research paper, thesis or class notes, can require a lot of mathematics. This can lead to many images. `LATEX2HTML` automatically detects when `LATEX` code is essentially identical to that used for an image already occurring within the document. A single image serves all such instances. However, even with this ‘image reuse’ the total number of images can still be large, giving significant loading delays.

math extension. One way to reduce these effects is to create *more* images, but of smaller pieces of mathematics. The idea is to extend the ‘simple math’ idea to use the text-font whenever possible. Only when a symbol or sub-expression cannot be represented adequately using the text-font is an image made. Any given HTML page can be expected to contain more images this way, however the same image may occur in many places on that page. The total size (in bytes) needed for images is reduced significantly, compared to when images are made of complete expressions.

Typically the first page is slow to load, as the images are downloaded across the network. Later pages in the same document require less download time as most of the required images will have been cached locally by the browser, from being present within earlier pages.

To activate the extra processing required for this strategy one must load `LATEX2HTML`’s special `math` extension, as follows:

```
latex2html -no_math -html_version 3.2,math ...
```

UNICODE fonts. Further reduction in the number of images is obtained by presuming that the browser will provide at least limited support for the UNICODE font encoding⁹. In particular there should be support for Greek letters, both upper and lower-case, and some extra mathematical symbols.

To activate this, append the `unicode` extension to the `-html_version` command-line switch (don’t leave any spaces):

```
... -no_math -html_version 3.2,math,unicode ...
```

Compare figure 4 with figure 3 to see the effect. This strategy is not yet ideal; notice the different styles of ϵ with and without the overline accent in the lower equations. Use of `\varepsilon` within the `LATEX` source alleviates this discrepancy; alternatively it may become possible for browsers to render accented UNICODE characters.

⁹ This is the case with the most recent versions of the *Netscape Navigator* and Microsoft’s *Internet Explorer*



$$\phi(\lambda) = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} \exp(u \ln u + \lambda u) du \quad \text{for } c \geq 0 \tag{1}$$

$$\lambda = \frac{\epsilon - \bar{\epsilon}}{\xi} - \gamma' - \beta^2 - \ln \frac{\xi}{E_{\max}} \tag{2}$$

$$\gamma = 0.577215... \quad (\text{Euler's constant}) \tag{3}$$

Figure 3: With the math extension loaded extra parsing of mathematics produces a mix of font-characters and smaller images. Opaque image backgrounds are used here only to show clearly which parts are images. In normal use these backgrounds are transparent.

$$\phi(\lambda) = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} \exp(u \ln u + \lambda u) du \quad \text{for } c \geq 0 \tag{1}$$

$$\lambda = \frac{\epsilon - \bar{\epsilon}}{\xi} - \gamma' - \beta^2 - \ln \frac{\xi}{E_{\max}} \tag{2}$$

$$\gamma = 0.577215... \quad (\text{Euler's constant}) \tag{3}$$

$$\gamma' = 0.422784... = 1 - \gamma \tag{4}$$

$$\bar{\epsilon} = \text{average energy loss} \tag{5}$$

$$\epsilon = \text{actual energy loss} \tag{6}$$

Figure 4: When the unicode extension is also loaded, Greek letters and other symbols can use font-characters also. This requires the browser to have some support for UNICODE.

Browser inadequacies. The commonly available Web browsers are continually improving, as more of the HTML 4.0 recommendations[7] are implemented. However some aspects of less advanced effects still create difficulties.

Look at the placement of superscripts and subscripts within figure 6. In mathematics these should be positioned above one another, as in B_{ij}^α . Furthermore the browser places extra space after italicized text. This is clearly evident in figure 5.

$$B_{ij}^\alpha = (B_{ij}^\alpha)_0 + (B_{ij}^\alpha)_a$$

$$(B_{ij}^\alpha)_0 = \frac{1}{2} \left(\frac{\partial N_i^\alpha}{\partial X_j} + \frac{\partial N_j^\alpha}{\partial X_i} \right)$$

$$(B_{ij}^\alpha)_a = H_{ij}^{\alpha\beta} \beta$$

Figure 5: A browser's placement of multiple superscripts and subscripts is not always ideal for mathematical usage.

$$V_{\pi^\alpha} = \left\langle \sum_i M_i V_i V_i + \sum_i \sum_{j>i} R_{ij} F_{ij} \right\rangle$$

$$= \left\langle \sum_i M_i V_i V_i + \sum_i \sum_{j>i} \sum_\alpha \sum_\beta r_{\alpha j \beta} f_{\alpha j \beta} - \sum_i \sum_\alpha p_{\alpha} f_{\alpha} \right\rangle$$

Figure 6: Some browsers place extra space after italicized text, over-compensating for the slope. This is particularly awkward for placing subscripts.

Such details should be fixed in future releases of browser software. Alternatively it may become possible to overcome these deficiencies within the HTML code, by specifying 'box-like' placement properties with a CSS style-sheet[7]. This requires browsers to support these advanced features.

Future support for MathML. As support for the new XML[7] (similar to HTML but more versatile) is incorporated into Web-browsers, it will become appropriate to extend the translation capabilities of L^AT_EX₂HTML. In particular, an ability to prepare

Bibliography

- 1 Белькович, А.А. රුසියානු ශිෂ්‍ය-භාෂා ශබ්දකෝෂය (*Русско-Сингальский Словарь*), Русский Язык, 1983.
- 2 Clough, Rev. B., සිංහල ඉංග්‍රීසි අකාරාදිය (*Sinhalese-English Dictionary*), Wesleyan Mission Press, Kollupitiya, 1892, facsimile edition by Asian Educational Services, New Delhi, 1982.

Figure 7: Bibliography entries using images for text of non-Latin based alphabets.

mathematics according to the MathML[7] markup scheme is a goal for future development.

Multi-lingual documents

Representing different languages within the same Web document presents problems similar to those with mathematics. There is no real difficulty when the languages are all based on the latin alphabet, provided any required accented letters are all available within a single font encoding.

The ISO-8859 encodings contain complete character-sets for various languages. Modern browsers provide support for Web pages having some of these as the designated character-set. \LaTeX 2HTML has specific support to produce pages using Latin-1, . . . , Latin-6 (i.e. ISO-8859-1, 2, 3, 4, 9 and 10). A particular character set is specified using an extension to the `\html_version` command-line switch. This is fully compatible with other extensions; e.g.

```
... -html_version 3.2,latin2,math,unicode ...
```

Images of special fonts. A single encoding rarely suffices when non-Latin languages are also required. Using images is a convenient strategy. Figure 7 shows some bibliographic entries¹⁰ using characters from the cyrillic alphabet and sinhalese script.

\LaTeX 2HTML recognises \TeX 's `\font` command as declaring a macro that will require an image to be made of enclosing environment. For example the cyrillic text was produced using:

```
\font\wncyr = wncyr at 10pt
...
...
{\wncyr Bel\char126koviq, A.A.} ...
```

Pre-processing for exotic scripts. The sinhalese script in figure 7 was generated in a similar way

to the cyrillic, but only *after* the source is filtered through Haralambous' *Indica* preprocessor, part of 'Sinhala- \TeX '[3]. After pre-processing, the \LaTeX source contains parts like:

```
{\SHb\char29a\char8}{\SHb\-\char69i}{\SHb...
```

in which each grouping generates an image for the appropriate letter or syllable. This is acceptable for small pieces of text in the exotic script. However many images are needed when there are whole paragraphs and pages of the script.

Automatic pre-processing. In figure 8 we see a portion in which each paragraph is presented as a separate image. One way is to use the `makeimage` environment, as was done with mathematics.

A better way is to use \LaTeX 2HTML on the manuscript, *before* pre-processing with *Indica*. Since the alphabets do not map one-for-one with the latin alphabet, a transliteration or transcription scheme is employed. Multi-letter combinations correspond to single letters or syllables in the exotic language. Portions of the manuscript using such schemes are included with the other parts to be rendered as images, just as with pieces of mathematics. The difference is that these portions need not be valid \TeX code, requiring pre-processing first. This is done as an extra step prior to image-generation.

Systems have been devised for the typesetting of various languages using \TeX , after first using such a pre-processing step. A suite of packages for \LaTeX and appropriate implementations for \LaTeX 2HTML, known as *Indic \TeX /HTML*[4], automate this process with some of the pre-processors available for Indic languages and traditional scripts. This includes support for Avinash Chopde's 'ITRANS' preprocessor [5] which handles many different languages and transliteration schemes.

The pre-processor was used this way for the page from which figure 8 was extracted. Some of

¹⁰ These are taken from a \LaTeX 2HTML conversion of the 'Sinhala- \TeX ' documentation[3], available at: <http://www-texdev.mpce.mq.edu.au/12h/indic/Sinhala/lreport/>

the HTML coding is shown in figure 9. Notice how the original transliteration is included as a comment. Just as with mathematics, this ensures the information is available even when the image fails to render.

UNICODE fonts, Ω and Λ . As UNICODE becomes more widely used, it should become possible to use its extensive range of characters, instead of images. Furthermore, it should become possible to employ Ω [2][6], via its \LaTeX variant Λ , in conjunction with \LaTeX 2HTML. It could be used for several tasks:

- as the pre-processing engine;
- replacing \LaTeX for the typesetting necessary when producing images;
- to generate UNICODE font-entities.

References

- [1] Nikos Drakos & Ross Moore, “The \LaTeX 2HTML Translator”. Documentation and User Guide accompanying the software; online version at <http://www-dsed.llnl.gov/files/programs/unix/latex2html/manual>.
- [2] Yannis Haralambous & John Plaice, “ Ω Times and Ω Helvetica Fonts Under Development: Step One”, *TUGboat*, The Communications of the \TeX Users Group, Volume 17, No. 2 (1996) pp. 126–146.
- [3] Yannis Haralambous & Dominik Wujastyk, “A Sinhalese \TeX System”, documentation for ‘Sinhala- \TeX ’ and the *Indica* preprocessor, 1994; available at <http://ctan.tug.org/ctan/tex-archive/languages/sinhala/>.
- [4] Ross Moore, “Indic \TeX /HTML, Traditional Scripts within Web-pages”, to appear in: *TUG-India*, volume 1, 1998; online version available at <http://www-texdev.mpce.mq.edu.au/indic/IndicHTML/>.
- [5] Avinash Chopde, *ITRANS* “Indian Language Transliteration Package”, A package for printing text in Indian Language Scripts, available from <http://www.aczone.com/itrans/>.
- [6] John Plaice & Yannis Haralambous, “The Latest Developments in Ω ”, *TUGboat*, The Communications of the \TeX Users Group, Volume 17, No. 2 (1996) pp. 181–183.
- [7] World Wide Web Consortium, online site at <http://www.w3c.org/Consortium/>; HTML 4.0: <http://www.w3c.org/Markup/> Stylesheets: <http://www.w3c.org/Style/> MathML: <http://www.w3c.org/Math/> XML: <http://www.w3c.org/XML/>



“හැබැද පුතා මෙහෙට මාරු උනා කියන්නේ?” නේ කෝප්පයන්
 තලගුලි තසීමන් රැගෙන ඉස්තෝප්පුවට ගිය සුදුහාමිනේ ඇසුවාය.
 “ඔව් ජනවාරියේ ඉදලා”
 “මේ දෙසැම්බර් මාසෙ. එතකොට ලබන මාසෙ ඉදලා”
 “ඔව් අලුත් වාරෙට”
 “කොහාටද මාරුව?”
 සුදුහාමිනේ එය අසන්ම වීරසේකර “හැබැට” කියා ඔව්ව ගැස්සුවේ එය
 අහන්නට තමාට අමතක වූ හෙයිනි.

Figure 8: Single images are made of whole paragraphs, when pre-processing is delayed until the image-generation phase.

```
<!-- INDICA S
  ‘e~ka nambuyine putha~’ vi~rase~kara katha~va patan gaththe~ nodhannekuta
  yamak kiya~ dhena paridhdeni.
  -->
<P><IMG
  WIDTH="554" HEIGHT="45" ALIGN="BOTTOM" BORDER="0"
  SRC="img4.gif"
  ALT="\lq\lq e~ka nambuyine putha~’ vi~rase~kara katha~va patan gaththe~ nodhannekuta
  yamak kiya~ dhena paridhdeni."></P>

<!-- INDICA S
  ‘e~ka nambuyi. koLa"mba ugannanava kiyandath puLuvan. i~tath koLa"mba loku
  isko~lavalata enne loku lokkange Lamayi. e~ Lamayi thama~ issarahata ho"ndha
  tha~nakata enne. i~tath e~ Lamayinge ma~rgayen puLuvan e~ Lamayinge
  tha~ththalagen o~na~ va~dak karava ganna~’
  -->
<P><IMG
  WIDTH="558" HEIGHT="108" ALIGN="BOTTOM" BORDER="0"
  SRC="img5.gif"
  ALT="\lq\lq e~ka nambuyi. koLa’’mba ugannanava kiyandath puLuvan. i~tath koLa’’mba lo
  ku...
  ..ma~rgayen puLuvan e~ Lamayinge tha~ththalagen o~na~ va~dak karava ganna~’’></P>

<!-- INDICA S
  ‘ballata dha~mu. ballata. ballata~’ baladhe~va sina~suNe~ya.
  -->
<P><IMG
  WIDTH="433" HEIGHT="22" ALIGN="BOTTOM" BORDER="0"
  SRC="img6.gif"
  ALT="\lq\lq ballata dha~mu. ballata. ballata~’ baladhe~va sina~suNe~ya."></P>
```

Figure 9: HTML code produced for some of the paragraphs of Sinhalese shown in figure 8, using a standard transliteration and preprocessed by *Indica*.

L^AT_EX code for figure 1

The following L^AT_EX code is adapted from pieces of coding provided by Michael Hall¹¹ and Michel Goossens,¹² for testing during the development of certain aspects of the mathematics support within L^AT_EX2HTML.

```
\documentclass[a4paper]{article}
\usepackage{html, amsmath, array, alltt}
\usepackage[dvips]{color}
% ensure \bm is defined if not latest LaTeX
\begin{latexonly}
\providecommand{\bm}[1]{\mathbf{#1}}
\end{latexonly}
\begin{imagesonly}
\providecommand{\bm}[1]{\mathbf{#1}}
\end{imagesonly}
\newcommand{\Range}{\mathcal{R}}
\newcommand{\Ker}{\mathcal{N}}
\newcommand{\Quat}{\vec{\mathbf{Q}}}
\renewcommand{\d}{\partial}

\begin{document}
\htmlhead[center]{section}{Math examples}

\begin{eqnarray}
\phi(\lambda) &= & \frac{1}{2} \pi \int_{-c}^c \ln |u + \lambda u| du \quad c \geq 0 \\
\lambda &= & \frac{\epsilon}{|\epsilon|} \gamma - \beta^2 \ln \frac{\xi}{E_{\max}} \\
\gamma &= & 0.577215 \dots \quad \text{(Euler's constant)} \\
\gamma' &= & 0.422784 \dots = 1 - \gamma \\
|\epsilon| &= & \text{average energy loss} \\
\epsilon &= & \text{actual energy loss}
\end{eqnarray}
```

Since $\delta \mathbf{c}$ or \mathbf{g} should hold for arbitrary \mathbf{c} \mathbf{v} -vectors, it is clear that $\mathbf{Ker}(A) = \mathbf{Range}(B)$ and that when $\mathbf{y} = B(\mathbf{x})$ then one has ...

```
\begin{eqnarray}\label{eqn:stress-sr}
V \mathbf{\pi}^{sr} &= & \left\langle \sum_i M_i \mathbf{V}_i \mathbf{V}_i + \sum_i \sum_{j>i} \mathbf{R}_{ij} \mathbf{F}_{ij} \right\rangle \\
\text{nonumber } &= & \left\langle \sum_i M_i \mathbf{V}_i \mathbf{V}_i + \sum_i \sum_{j>i} \sum_{\alpha} \sum_{\beta} \mathbf{r}_{i\alpha} \mathbf{j}_{\beta} \mathbf{f}_{i\alpha} \mathbf{j}_{\beta} - \sum_i \sum_{\alpha} \mathbf{p}_{i\alpha} \mathbf{f}_{i\alpha} \right\rangle
\end{eqnarray}

\begin{subequations}\label{bgdefs}
\begin{align}
B_{ij}^{\alpha} &= \left( B_{ij}^{\alpha} \right)_0 + \left( B_{ij}^{\alpha} \right)_a \quad \text{\label{bdef}} \\
\left( B_{ij}^{\alpha} \right)_0 &= \frac{1}{2} \left( \frac{\mathbf{d} N_i^{\alpha}}{\mathbf{d} X_j} + \frac{\mathbf{d} N_j^{\alpha}}{\mathbf{d} X_i} \right) \quad \text{\label{b0def}} \\
\left( B_{ij}^{\alpha} \right)_a &= H_{ij}^{\alpha \beta} a^{\beta} \quad \text{\label{budef}} \\
H_{ij}^{\alpha \beta} &= \frac{1}{2} \left( \frac{\mathbf{d} N_k^{\alpha}}{\mathbf{d} X_i} \frac{\mathbf{d} N_k^{\beta}}{\mathbf{d} X_j} + \frac{\mathbf{d} N_k^{\beta}}{\mathbf{d} X_i} \frac{\mathbf{d} N_k^{\alpha}}{\mathbf{d} X_j} \right) \quad \text{\label{gdef}}
\end{align}
\end{subequations}

\end{document}
```

¹¹ Dr. Michael L. Hall, Los Alamos National Laboratory.

¹² Dr. Michel Goossens, IT Division, CERN, Geneva