dynMath: A PostScript Type 3-based LaTeX package to support extensible mathematical symbols

Abdelouahad Bayar

Abstract

This paper gives an overview of the characteristics and capabilities of a package called dynMath. The main aim of this package is to provide LaTeX with the ability to support dynamic mathematical symbols, thereby improving the quality of the scientific document. Dynamic mathematical symbols are developed in such a way that, when stretched, they will respect *optical scaling*, *uniformity* of shape, *right-sizing* and the closest possible *likeness* to their *analogues* in the *old printing system* (metal typesetting). The tools supplying the stretching with the above characteristics are supported in part by LaTeX (dynMath) and in part by a PostScript Type 3 font.

1 Introduction

In a scientific document, mathematical formulas are written using static and/or variable-sized symbols. In a document typeset at a given size, the shape and the size of static symbols remain unchanged throughout the document. A variable-sized symbol or simply a dynamic symbol varies in terms of size and sometimes shape from one context to another in the same document. The formula in Figure 1, typeset with (normal) LaTeX, is referenced to clarify the concept. The symbols A, a, B, b, C and c are static symbols whereas the parentheses delimiters are dynamic. We notice that the parentheses change in dimension in order to cover the formula to be delimited. In addition, even the shape has changed.

Without a doubt, (LA)TeX is the most widely used system by the scientific community in typesetting scientific documents. (LA)TeX supports the concept of dynamic symbols or operators in all its different implementations, i.e. TeX [8], LaTeX [10], LuaTeX [12], . . .

Considering Figure 1, the parentheses delimiting matrix C are not of the same form as those in the case of matrices A and B. So, with (LA)TeX, we lose in *uniformity* of the shape of the parentheses when the height of the formula exceeds a certain level. This is not the case in Figure 2, where the same formulas are reproduced with dynMath.

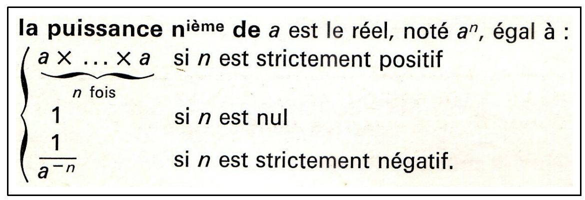
Some of the variable-sized mathematical symbol shapes used in the old printing system (metal typesetting) are supported in digital printing, but only with close shapes, for technical reasons. This causes a slight drop in quality (beauty). The case is illustrated by considering an extract from an old mathematics book printed in 1974 [13], as shown in Figure 3. The symbol concerned is the brace. To illustrate the idea, the book's script is reproduced with LaTeX as shown in Figure 4. The property of *resemblance* to analogous metal symbols will be called "*metal-likeness*".

$$A = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}, B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}, C = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ c_n \end{pmatrix}$$

Figure 1: Variable-sized parentheses with LaTeX

$$A = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}, B = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}, C = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ c_n \end{pmatrix}$$

Figure 2: Variable-sized parentheses with dynMath


la puissance n$^{\text{ième}}$ de a est le réel, noté a^n, égal à :

$$\begin{cases} \underbrace{a \times \ldots \times a}_{n \text{ fois}} & \text{si } n \text{ est strictement positif} \\ 1 & \text{si } n \text{ est nul} \\ \frac{1}{a^{-n}} & \text{si } n \text{ est strictement négatif.} \end{cases}$$

Figure 3: Variable-sized braces in metal typesetting

La puissance n$^{\text{ième}}$ de a est le réel, noté a^n, égal à :

$$\begin{cases} \underbrace{a \times \ldots \times a}_{n \text{ fois}} & \text{si } n \text{ est strictement positif} \\ 1 & \text{si } n \text{ est nul} \\ \frac{1}{a^{-n}} & \text{si } n \text{ est strictement négatif} \end{cases}$$

Figure 4: Variable-sized braces with LaTeX

Referring once again to Figure 3, we can see that the two braces are not related by a linear relation. The ratio between heights is not the same as for thicknesses. The scaling model used in document processing (old and current) is not linear. In document processing, this is called *"optical scaling"*. More details on the optical scaling concept are found in [2, 5, 6]. In LaTeX, the scaling used is not linear, though it's also not quite optical, since after a certain size, the thickness becomes constant.

Extensible delimiter symbols in (LA)TeX, when the size of the formula to be delimited exceeds a known level, are obtained by a composition based on more than one character. One of the component characters is repeated as many times as necessary

Abdelouahad Bayar

doi.org/10.47397/tb/45-1/tb139bayar-dynmath

until the size of the delimiter matches, not exactly, but approximately the suggested size. (LA)TEX lacks the property that we'll call "*right-sizing*".

Properties such as uniformity, optical scaling, metal-likeness and right-sizing, which are missing in normal TEX, are principally brought by `dynMath`.

The support of dynamic characters, particularly mathematical symbols, has been a topic of research in the field of electronic document processing for over four decades. It has focused on the extensible Arabic letter to supply the concept of the Kashida, the basis of justification of Arabic scripts [1, 4, 7, 11] especially as it has concerned dynamic mathematical symbols [2, 11].

`CurExt` is a LATEX extension developed to provide the possibility of typesetting mathematical formulas using variable-sized symbols. It supports extensible parentheses and the Kashida, the mechanism needed to extend Arabic mathematical symbols. The `CurExt` package does not offer the ability to manipulate braces and other notations with metal-likeness. Its development principle, taking into account its ability to handle parentheses, enables it, once completed, to meet our four properties of uniformity, optical scaling, metal-likeness and right-sizing.

We should point out that the `dynMath` package is the direct continuation of research published before [3]. The background mathematical model supporting stretching and the TEX macros developed in [3] has been revised and improved to supply as well as possible uniformity, optical scaling, metal-likeness and right-sizing.

The remainder of this paper is organized as follows. In Section 2, we give a brief description of the general layout relative to the `dynMath` package. In section 3, the metal-likeness of dynamic symbols in `dynMath` is defined and discussed. The concepts of "uniformity" and "right-sizing" are studied together in Section 4. In Section 5, the ability to support some old mathematical symbolic notations is presented. In Section 6, the general mode to build the naming of macros to be used by the typesetter is highlighted. Since the paper is a presentation of a LATEX package, some samples in terms of source codes and formatting results are given in Section 7. The paper ends with conclusions and perspectives.

2 General layout of `dynMath`

The `dynMath` package has the ability to support dynamic mathematical symbols due to the existing possibilities of interaction between (LA)TEX and PostScript. This is done using the command `\special` via the `dvips` driver to translate `dvi` files to PostScript [14]. More precisely, we use these methods to include literal PostScript in TEX documents to work with the PostScript Type 3 font. Thus, to format a LATEX source file named `doc.tex` for example, and get the corresponding PDF, the command line needs to be (or the equivalent for LuaLATEX):

```
latex doc.tex; dvips doc.dvi; \
ps2pdf doc.ps
```

At this time, the `dynMath` package consists of two files: `dynMath.sty` and `dynMath.tps`. These two files must be added to the TEX distribution or to the working directory. A brief description of these two files is given below.

- `dynMath.sty`: contains the LATEX package specification in TEX and LATEX programming. It contains all useful variables, macros and interfaces with the content of the `dynMath.tps` file.
- `dynMath.tps`: contains a PostScript Type 3 font named `dynMath` as a literal header that is a '`!`' `\special`. The content of the file looks like : `\special {!...` ⟨*dynMath specification*⟩ `...}`. This is PostScript code inside TEX source, for which we adopted the extension `.tps` to mean TEX (t) and PostScript (ps).

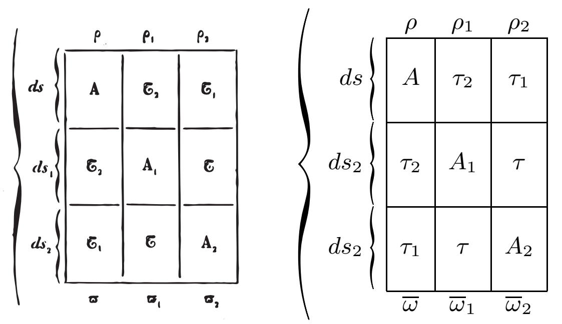
3 Metal-likeness in `dynMath`

The main goal of the research work is the ability to support dynamic characters in document processing. But one of the principal motivations is to add to current document processing tools the capability of supplying stretchable mathematical symbols looking like those printed in metal typesetting. The LATEX package `dynMath` brings this opportunity. In fact, braces supplied by `dynMath` for example (which are among the most difficult symbols to support) look like those from metal typesetting. We call this concept "*metal-likeness*".

An illustration is shown in Figure 5. We consider a mathematical presentation taken from an old book [9] (Figure 5a) and we typeset it using `dynMath` (Figure 5b). There's a difference in the script fonts in the mathematical equation, but this doesn't matter since it's the braces we're interested in.

4 Uniformity and right-sizing

`dynMath` supports dynamic mathematical symbols. It also has the ability to handle these symbols in their exact and tailored sizes or in any size convenient to a context. Let us consider the parenthesis as an example of a dynamic symbol to introduce the concept. We need first to recall in an abstract way how documents are printed through (LA)TEX systems. When printing a document formatted with



(a) Metal left brace symbols

(b) `dynMath` left brace symbols

Figure 5: Metal and `dynMath` symbols comparison

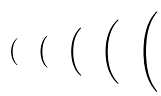


Figure 6: Standalone left parentheses

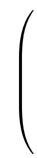
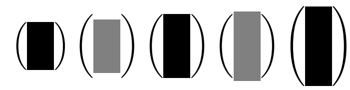
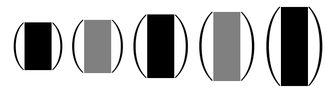


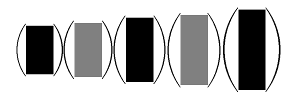
Figure 7: A left compound parenthesis



(a) Approximated (LA)TEX delimitation



(b) Exact `dynMath` delimitation



(c) Non-approximated `CurExt` delimitation

Figure 8: Approximated versus exact delimitations

(LA)TEX, the bitmaps of the characters are used instead of the METAFONT language encoding. It is well-known that applying scaling operations to bitmaps diminishes the quality of the images. Consequently, the support of dynamic fonts based directly on bitmaps is not a feasible way to print documents in good quality.

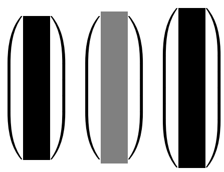
This problem has already been solved by D. E. Knuth. The approach is presented considering the parenthesis. Knuth has designed standalone parentheses (not composed from other characters) of different sizes as shown in Figure 6. When the height of a mathematical formula is less than the highest of these five parentheses (see Figure 6), the closest parenthesis in terms of size is used to delimit the formula.

The previous idea can not be used to solve entirely the problem since it is necessary to give a large number of parentheses in different sizes to cover all the needs. In addition, we can not predict in a meaningful way a maximum size of mathematical formulas. When a parenthesis, of a height exceeding the highest standalone parenthesis, is needed, (LA)TEX uses a compound parenthesis based on three characters: ⎛, ⎝ and ⎜. The last is repeated as many times as necessary between the first and the second to optimally cover the formula to be delimited. A sample of a compound parenthesis is shown in Figure 7. Two important things to note. The first is that the parenthesis used, whether standalone or compound, will not always be exactly of the same height as the formula to delimit. The second relates particularly to the compound parenthesis: it differs from the first five ones in terms of shape. This leads therefore to a loss of *uniformity*. Let us notice that the symbols in Figure 6 are true parentheses.

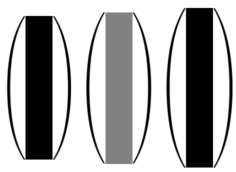
Here, we have to point out that `CurExt`, like `dynMath`, also handles the sizes of parentheses in a precise and not approximate way. One difference between `dynMath` and `CurExt` is that in the latter the parentheses' sizes are a little larger than the real size of the formula in order to cover it integrally. This possibility will be supported in the future by `dynMath` as an option.

`dynMath`, developed on the basis of a mathematical model supporting the optical scaling concept, allows for parentheses of sizes satisfying exactly the needs of a formula, using PostScript Type 3 fonts. Even in the case of large formulas, the delimiter produced by `dynMath` retains the form of a parenthesis ensuring the keeping of the "*uniformity*" property. Once again, it must be mentioned that `CurExt` supports these characteristics.

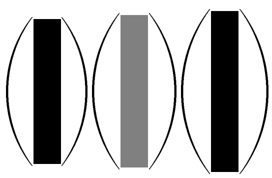
The concept is clarified considering the abstract formulas delimited by parentheses using normal TEX, `dynMath` and `CurExt` in Figures 8a, 8b and 8c. We notice that the abstract mathematical formulas modeled by black boxes in Figure 8a are of sizes equal to existing parentheses in the `cmex10` font. This is why the delimiting parentheses are exactly of the same size (height plus depth). But in the case of gray formulas, only the nearest parenthesis in terms of size is used. With `dynMath`, in all cases the parentheses are exactly of the same size as the formulas to delimit. About `CurExt`, the exact size of the formula is taken and increased a little to determine the



(a) Approximated (LA)TEX delimitation, losing uniformity



(b) Exact dynMath delimitation, keeping uniformity



(c) Non-approximated CurExt delimitation, keeping uniformity

Figure 9: Approximated delimitation losing uniformity, versus exact delimitation keeping uniformity

(a) Constant (approximated) (LA)TEX delimitation

(b) Exact dynMath delimitation

(c) Non-approximated CurExt delimitation

Figure 10: Approximated versus exact delimitation for global height less than 10pt

size of the parentheses. In this way, it is an exact delimitation or simply a non-approximated one.

Figures 9a, 9b and 9c present the case where the size of a formula exceeds that of the standalone parentheses. Compound parentheses are used with (LA)TEX. As before, (LA)TEX uses approximation in the delimitation process, in contrast to dynMath and CurExt. In addition, with (LA)TEX, the delimiters lose the real shape of parentheses and so the *uniformity* of delimitation. With dynMath and CurExt, the general shape is kept.

It's important to mention the case where the overall height (height + depth) of the formula to be delimited is less than 10pt. As shown in Figure 10a, (LA)TEX uses the smallest standalone parenthesis to delimit mathematical formulas in this case. In a way, this is also a case of approximation. On the other hand, dynMath and CurExt use adapted parentheses (see Figures 10b and 10c).

We can see that the dynMath package comes to extend (LA)TEX's capabilities. When processing a document, typesetters can use dynMath and the mathematical capabilities of LATEX at the same time

$$n^p = \overbrace{n \times n \times \ldots \times n}^{p\,\text{times}}$$

(a) (LA)TEX over bracing

$$n^p = \overbrace{n \times n \times \ldots \times n}^{p\,\text{times}}$$

(b) dynMath over bracing

$$n^p = \underbrace{n \times n \times \ldots \times n}_{p\,\text{times}}$$

(c) (LA)TEX under bracing

$$n^p = \underbrace{n \times n \times \ldots \times n}_{p\,\text{times}}$$

(d) dynMath under bracing

$$\dot{\overline{n_1 + n_2 + \ldots + n_p}}$$

(e) (LA)TEX equivalence classes

$$\dot{\overset{\frown}{n_1 + n_2 + \ldots + n_p}}$$

(f) dynMath equivalence classes

Figure 11: (LA)TEX versus dynMath decorations

without any problem. In designing dynMath, we insisted on retaining what is common to dynMath and normal LATEX, staying faithfully as possible to LATEX, and at the same time adding what can't be supported by LATEX. In Figure 8, we can observe that the thickness, the spaces separating the delimiters and the formulas, in the cases where the size of the formulas coincides with the size of a particular standalone parenthesis, are the same. Otherwise, dynMath adds what couldn't be achieved with LATEX in terms of size/shape of parentheses (see Figures 8 and 9).

5 Support of some old mathematical symbolic notations

With advancement in computer technology and software tools, (scientific) document processing has seen great improvements in terms of time, flexibility of processing, quality and so on. But, for technical reasons, many symbols and notations have been replaced by others or completely forsaken. These notations or symbols are the foundation of quality in book typesetting in traditional printing. Parentheses and braces are good examples. With dynMath, LATEX can do more of these lost things. We will not show again these two symbols in the delimitation case, but in decoration or diacritics. We consider the LATEX macros \overbrace (see Figure 11a), \underbrace (see Figure 11c), \overline and \dot (see Figure 11e) to show more of what dynMath can add to LATEX's abilities. The corresponding possibilities supported by dynMath are given in Figures 11b, 11d and 11f. With dynMath, we can now work in the old way to define equivalence classes, specifying them with dotted parentheses and not just dotted lines (see Figure 11f).

6 Global syntax mode

In dynMath, the delimitation of formulas is done with the macro \meLeft...\meRight. We followed the command names \left...\right but started with a capital letter and preceded by "me" meaning metal (referencing the metal symbols). The macros operates in the same way as \left...\right, not neglecting the fact that the parameters (formulas) must be enclosed between braces (indicating group) when the formula contains more than one token.

We have followed this way in order to make using dynMath especially easy for users that are familiar with typesetting mathematical formulas under (LA)TEX. For example, to handle braces over formulas we programmed the macro \meOverBrace of the name coming from the (LA)TEX macro \overbrace. We will follow this method when implementing what remains to complete the package.

7 Samples

In this section, we will present some samples of formulas formatted with dynMath, also giving the corresponding LATEX. In particular, the code of some examples used in the previous section will be considered in the current one.

dynMath requires the package mathstyle and so the latter must be installed in the (LA)TEX distribution used. We recall that the PostScript Type 3 font used as the base of dynMath is provided in the file dynMath.tps. Thus, both dynMath.sty and dynMath.tps also have to be added to the (LA)TEX distribution or copied into the directory containing the LATEX file to format.

The document to format using dynMath must contain the usual line \usepackage{dynMath}.

We will give some samples of LATEX sources based on dynMath followed by the corresponding generated output. To compare with normal LATEX, we use the same source code with \left and \right instead of \meLeft and \meRight respectively.

7.1 Sample one: parentheses

dynMath source:

```
\[ \meLeft({
\begin{array}{cc}
\meLeft({
\begin{array}{cc}
a_{11}& a_{12}\\a_{21}& a_{22}\\a_{31}& a_{32}
\end{array}
}\meRight)
& ... bij components  \\
  ... cij components & ... dij components
}\meRight)
\end{array}
}\meRight) \]
```

dynMath corresponding output:

$$\left(\begin{array}{cc}\left(\begin{array}{cc}a_{11} & a_{12}\\ a_{21} & a_{22}\\ a_{31} & a_{32}\end{array}\right) & \left(\begin{array}{cc}b_{11} & b_{12}\\ b_{21} & b_{22}\\ b_{31} & b_{32}\end{array}\right)\\ \left(\begin{array}{cc}c_{11} & c_{12}\\ c_{21} & c_{22}\\ c_{31} & c_{32}\end{array}\right) & \left(\begin{array}{cc}d_{11} & d_{12}\\ d_{21} & d_{22}\\ d_{31} & d_{32}\end{array}\right)\end{array}\right)$$

LATEX source:

```
\[ \left({
\begin{array}{cc}
...
\end{array}
}\right) \]
```

LATEX corresponding output:

$$\left(\begin{array}{cc}\left(\begin{array}{cc}a_{11} & a_{12}\\ a_{21} & a_{22}\\ a_{31} & a_{32}\end{array}\right) & \left(\begin{array}{cc}b_{11} & b_{12}\\ b_{21} & b_{22}\\ b_{31} & b_{32}\end{array}\right)\\ \left(\begin{array}{cc}c_{11} & c_{12}\\ c_{21} & c_{22}\\ c_{31} & c_{32}\end{array}\right) & \left(\begin{array}{cc}d_{11} & d_{12}\\ d_{21} & d_{22}\\ d_{31} & d_{32}\end{array}\right)\end{array}\right)$$

7.2 Sample two: braces

dynMath source:

```
\[
\meLeft\{{
 \begin{array}{l}
  \meLeft\{{
   \begin{array}{lcl}
    f_{1}\meLeft({x_{1},x_{2},\cdots,x_{9}}
           \meRight) & = & F_{1}\\
    f_{2}\meLeft({x_{1},x_{2},\cdots,x_{9}}
           \meRight) & = & F_{2}\\
    f_{3}\meLeft({x_{1},x_{2},\cdots,x_{9}}
           \meRight) & = & F_{3}
   \end{array}
  }\meRight.\\
  \\
  \meLeft\{{
   \begin{array}{lcl}
    ... gi equations
   \end{array}
  }\meRight.\\
  \\
  \meLeft\{{
   \begin{array}{lcl}
    ... hi equations
   \end{array}
  }\meRight.
 \end{array}
}\meRight.
\]
```

dynMath corresponding output:

$$
\left\{ \begin{array}{l}
\left\{ \begin{array}{lcl}
f_1(x_1, x_2, \cdots, x_9) & = & F_1 \\
f_2(x_1, x_2, \cdots, x_9) & = & F_2 \\
f_3(x_1, x_2, \cdots, x_9) & = & F_3
\end{array} \right. \\
\\
\left\{ \begin{array}{lcl}
g_1(x_1, x_2, \cdots, x_9) & = & G_1 \\
g_2(x_1, x_2, \cdots, x_9) & = & G_2 \\
g_3(x_1, x_2, \cdots, x_9) & = & G_3
\end{array} \right. \\
\\
\left\{ \begin{array}{lcl}
h_1(x_1, x_2, \cdots, x_9) & = & H_1 \\
h_2(x_1, x_2, \cdots, x_9) & = & H_2 \\
h_3(x_1, x_2, \cdots, x_9) & = & H_3
\end{array} \right.
\end{array} \right.
$$

LaTeX source:

```
\[
\left\{{ \begin{array}{l}
...
 \end{array}
}\right. \]
```

LaTeX corresponding output:

$$
\left\{ \begin{array}{l}
\left\{ \begin{array}{lcl}
f_1\left(x_1, x_2, \cdots, x_9\right) & = & F_1 \\
f_2\left(x_1, x_2, \cdots, x_9\right) & = & F_2 \\
f_3\left(x_1, x_2, \cdots, x_9\right) & = & F_3
\end{array} \right. \\
\\
\left\{ \begin{array}{lcl}
g_1\left(x_1, x_2, \cdots, x_9\right) & = & G_1 \\
g_2\left(x_1, x_2, \cdots, x_9\right) & = & G_2 \\
g_3\left(x_1, x_2, \cdots, x_9\right) & = & G_3
\end{array} \right. \\
\\
\left\{ \begin{array}{lcl}
h_1\left(x_1, x_2, \cdots, x_9\right) & = & H_1 \\
h_2\left(x_1, x_2, \cdots, x_9\right) & = & H_2 \\
h_3\left(x_1, x_2, \cdots, x_9\right) & = & H_3
\end{array} \right.
\end{array} \right.
$$

7.3 Sample three: Figure 3

The mathematics in Figure 3 was typeset in native LaTeX in Figure 4. It remains to typeset it with dynMath, as follows. The dynMath source:

```
\begin{minipage}{7cm}
\textbf{La puissance n$^{\mbox{\scriptsize
  i\`{e}me}}$ de }$a$ est le réel,
  noté $a^{n}$,\\ égal~à~:\\
$
\meLeft\{%
  \begin{array}{ll}%
    \meUnderBrace{a\times\ldots\times a}
      _{n\textrm{ fois}} &
    \kern -3pt\textrm{si }n
      \textrm{ est strictement positif}\\
    1 & \kern -3pt\textrm{si }n
      \textrm{ est nul}\\
    \frac{1}{a^{-n}} & \kern -3pt\textrm{si }n
      \textrm{ est strictement négatif}
  \end{array}
\meRight.
$
\end{minipage}
```

dynMath corresponding output:

La puissance n$^{\textbf{ième}}$ de a est le réel, noté a^n, égal à :

$$
\left\{ \begin{array}{ll}
\underbrace{a \times \ldots \times a}_{n\ \mathrm{fois}} & \text{si } n \text{ est strictement positif} \\
1 & \text{si } n \text{ est nul} \\
\frac{1}{a^{-n}} & \text{si } n \text{ est strictement négatif}
\end{array} \right.
$$

7.4 Sample four: over brace

dynMath source:

```
\[ n^p=\meOverBrace{n\times n\times\ldots
    \times n}^{p\, \mathrm{times}} \]
```

dynMath corresponding output:

$$n^p = \overbrace{n \times n \times \ldots \times n}^{p\, \mathrm{times}}$$

LaTeX source:

```
\[ n^p=\overbrace{...} \]
```

LaTeX corresponding output:

$$n^p = \overbrace{n \times n \times \ldots \times n}^{p\, \mathrm{times}}$$

7.5 Sample five: under brace

dynMath source:

```
\[ n^p=\meUnderBrace{n\times n\times\ldots
    \times n}_{p\, \mathrm{times}} \]
```

dynMath corresponding output:

$$n^p = \underbrace{n \times n \times \ldots \times n}_{p\, \mathrm{times}}$$

LaTeX source:

```
\[ n^p=\underbrace{...} \]
```

LaTeX corresponding output:

$$n^p = \underbrace{n \times n \times \ldots \times n}_{p\, \mathrm{times}}$$

7.6 Sample six: over dot

dynMath **source:**

```
\[ \dot{\meOverParenthesis{n_{1}+n_{2}+
     \ldots + n_{p}}} \]
```

dynMath corresponding output:

$$\dot{\overparen{n_1 + n_2 + \ldots + n_p}}$$

LaTeX source:

```
\[ \dot{\overline{...}} \]
```

LaTeX corresponding output:

$$\dot{\overline{n_1 + n_2 + \ldots + n_p}}$$

So far, we've talked in terms of concepts and examples about parentheses and braces as dynamic symbols. The latter represent the category of symbols with extensible parts based on curvilinear (nonlinear) curves. They remain among the most difficult to support in `dynMath`. This is not the case for extensible symbols whose dynamic parts are purely lines, which are easy to parameterize for extension. Square brackets are a good example. The following sample shows the delimitation of a multi-matrix by square brackets.

7.7 Sample seven: square brackets

`dynMath`:

$$\begin{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} & \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \\ \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \\ c_{31} & c_{32} \end{bmatrix} & \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \\ d_{31} & d_{32} \end{bmatrix} \end{bmatrix}$$

LaTeX:

$$\begin{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} & \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \\ \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \\ c_{31} & c_{32} \end{bmatrix} & \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \\ d_{31} & d_{32} \end{bmatrix} \end{bmatrix}$$

The source for the above formulas is the same as in Sample 1, except that the third lines of the two high matrices are removed and the delimiters '(' and ')' are replaced by '[' and ']' respectively.

8 Conclusions

`dynMath` is a LaTeX package that supports dynamic mathematical symbols with respect to *optical scaling*, *metal-likeness*, *uniformity* and *right-sizing*. It will be finalized soon and deposited in the CTAN (Comprehensive TeX Archive Network) repository with a complete user manual. As the symbols are programmed in PostScript, they are not affected by the color effects controlled by LaTeX. As a future enhancement, `dynMath` will be extended to support color interaction with PostScript.

References

[1] A. Anane. Arabic text justification using LuaLaTeX and the DigitalKhatt OpenType variable font. *TUGboat* 42(3):247–257, 2021. `https://tug.org/TUGboat/tb42-3/tb132anane-variable.pdf`

[2] J. André, I. Vatton. Dynamic optical scaling and variable-sized characters. *Electronic Publishing*, 7(4):231–250, 1994. `https://jacques-andre.fr/japublis/opticalscaling.pdf`

[3] A. Bayar. Towards an operational (LA)TeX package supporting optical scaling of dynamic mathematical symbols. *TUGboat* 37(2):171–179, 2016. TUG 2016 (Toronto) conference proceedings. `https://tug.org/TUGboat/tb37-2/tb116bayar.pdf`

[4] D.M. Berry. Stretching letter and slanted-baseline formatting for Arabic, Hebrew, and Persian with ditroff/ffortid and dynamic PostScript fonts. *Software: Practice and Experience*, 29(15):1417–1457, 1999. `https://cs.uwaterloo.ca/~dberry/FTP_SITE/tech.reports/keshide.paper.pdf`

[5] Circuitous Root. Clubs and cults revisiting the concept of 'typeface' and the optical scale in typefounding, 2013. `https://www.circuitousroot.com/artifice/letters/press/typemaking/making-matrices/terms/logical-grouping/clubs-and-cults/index.html`

[6] Circuitous Root. From the optical scale to optical scaling, 2013. `https://www.circuitousroot.com/artifice/letters/press/typemaking/mats/optical/index.html`

[7] M. Elyaakoubi, A. Lazrek. Justify just or just justify. *Journal of Electronic Publishing*, 13(1), 2010.

[8] D.E. Knuth. *The TeXbook*. Addison-Wesley, Reading, Massachusetts, 1st ed., 1984.

[9] G. Lamé. *Leçons sur les coordonnées curvilignes et leurs diverses applications*. Imprimerie de Mallet Bachelier, Rue du Jardinet 12, Paris, 1859. `https://archive.org/details/leonssurlescoor01lamgoog`

[10] L. Lamport. *LaTeX — A Document Preparation System*. Addison Wesley, USA, 1994.

[11] A. Lazrek. CurExt, typesetting variable-sized curved symbols. *TUGboat* 24(3):323–327, 2003. EuroTeX 2003 (Brest) conference proceedings. `https://tug.org/TUGboat/tb24-3/lazrek.pdf`

[12] LuaTeX development team. *LuaTeX Reference Manual*. `https://ctan.org/pkg/luatex`

[13] M. Monge, M.C. Audouin-Egoroff, F. Lemaire-Body. *Arithmetiques, Analyse et Probabilités, Terminal C et E*. Librairie Classique Eugène Belin, France, 1974.

[14] T. Rokicki. *Dvips: A DVI-to-PostScript Translator*. `https://tug.org/dvips`

◇ Abdelouahad Bayar
Cadi Ayyad University — Higher School of Technology of Safi
Sidi Aissa Road, PB 89
Safi, 46000
Morocco
`a.bayar (at) uca dot ma`
ORCID 0000-0002-3496-505X