CALS TEST NETWORK

# CTN Test Report
# 93-019

# A User's Guide To MIL-M-28001B SGML Declaration For Mathematical Formulae

**22 April 1993**



**Prepared For**
**Air Force CALS Program Office**
**Det 2 HQ ESC/ENC**
**Wright-Patterson AFB OH 45431-1601**

CTN Report 93-019
22 APRIL 1993

---

# A USER'S GUIDE TO MIL-M-28001B SGML DECLARATIONS FOR MATHEMATICAL FORMULAE

22 April 1993

---

**Prepared by:**
Navy Test Bed, CALS Test Network
Carderock Division
Naval Surface Warfare Center
Bethesda, MD 20084-5000

**NSWC Contact:**
Donald Gignac
(301) 227-3348

Contents

## 1. INTRODUCTION

The following is a guide to the use of the 'mathpac' elements and entities provided in section 70 MATH DECLARATION SET of appendix C in either MIL-M-28001A or MIL-M-28001B. The 'mathpac' entities and elements are to be used in the SGML tagging of mathematical notation (equations, formulas, etc.) that occurs in documents conforming to MIL-M-28001. This 'mathpac' material is based on similar material in section 8 'Mathematics' of the technical report ISO/IEC TR 9573 'Information Processing - SGML Support Facilities - Techniques for Using SGML'. For the reader's convenience, a data dictionary of the 'mathpac' tags is included in an appendix to this guide.

Clearly some rudimentary knowledge of both mathematical notation and SGML is required to use the 'mathpac' tags. If necessary the author can assist with the tagging of complicated mathematical notation by answering questions with regard to unfamiliar symbols, their meaning, usage, etc. The author can also assist with the formatting of such notation by advising how it should be presented and arranged on the page.

With regard to SGML, a brief introduction is provided in section 30 INTRODUCTION of appendix A in either MIL-M-28001A or MIL-M-28001B. For more detailed information concerning SGML, the reader is referred to the bibliography supplied in Section 1.1.

This is a brief guide to the use of 'mathpac' tags in marking up mathematical notation in documents conforming to MIL-M-28001A or MIL-M-28001B. For additional examples and more detailed information concerning the 'mathpac' tags, the reader should consult the following sources:

a. the SGML comments preceding the declarations in section 70.1 'Math declaration set' of Appendix C in either MIL-M-28001A or MIL-M-28001B.

b. the explanatory material in subsections 8.1 through 8.6 of TR 9573, as well as the sample markups in the SGML comments preceding declarations in subsection 8.7 'Element and entity definition'.

However, the reader should note the following two points with regard to the sample markups provided in TR 9573. TR

9573 sometimes makes use of the 'shorttag' SGML feature which is not allowed in the MIL-M-28001B SGML environment. The 'shorttag' feature allows some minimization of notation in markup. Consequently, some TR 9573 'mathpac' examples will have to be 'translated' into MIL-M-28001B SGML.

For example, the markup 'x<sub/i/<sup/2/' will not parse using the MIL-M-28001 SGML Environment. The mark-up will have to be expanded to 'x<sub>i</sub><sup>2</sup>' by expanding the null end tag notation as follows. The '/' after 'i' and the '/' after '2' are NETs (null end tags). The first '/' takes the place of '</sub>' and the second '/' takes the place of '</sup>'. However, before this can happen each NET must be preceded by a NET-enabling start tag. A NET-enabling start tag is a start tag where the concluding '>' is replaced with '/'.

Also tags such as '<fence type=brace style=double>...' will have to be rewritten as '<fence type="brace" style= "double">...</fence>' by delimiting attribute values. The 'shorttag' feature allows delimiters to be omitted when the attribute value consists of name characters.

Moreover, the 'mathpac' declarations in TR 9573 include 'shortref' and their associated 'usemap' declarations. Short references are basically user-defined macros; their use facilitates the markup of complex SGML constructs such as equations and tables. However, short reference usage is not allowed in the MIL-M-28001B SGML environment (SHORTREF is NONE on the MIL-M-28001B SGML Declaration) since FIPS PUB 152 prohibits it.

## 1.1 BIBLIOGRAPHY

(1) 'Information processing - Text and office systems - Standard Generalized Markup Language (SGML)', ISO 8879, International Organization for Standards, Geneva, 1986.

(2) 'Information processing - Text and office systems - Standard Generalized Markup Language (SGML) Amendment 1', ISO 8879, International Organization for Standards, Geneva, 1988.

(3) 'Information processing - SGML support facilities - Techniques for using SGML', ISO/IEC TR 9573, International Organization for Standards, Geneva, 1988.

(4) 'MARKUP REQUIREMENTS AND GENERIC STYLE SPECIFICATION FOR ELECTRONIC PRINTED OUTPUT AND EXCHANGE OF TEXT', MIL-M-28001A, 20 July 1990.

(5) 'MARKUP REQUIREMENTS AND GENERIC STYLE SPECIFICATION FOR ELECTRONIC PRINTED OUTPUT AND EXCHANGE OF TEXT', MIL-M-28001B (draft), 13 December 1991.

(6) 'SGML: The User's Guide to ISO 8879' by Joan M. Smith and Robert Stutely, John Wiley, 1988.

(7) 'The SGML Handbook' by Charles M. Goldfarb, Oxford University Press, 1990.

(8) 'SGML An Author's Guide to the Standard Generalized Markup Language' by Martin Bryan, Addison-Wesley, 1988.

(9) 'Practical SGML' by Eric van Herwijnen, Kluwer Academic Publishers, 1990.

(10) 'Standard Generalized Markup Language (SGML)', Federal Information Processing Standards Publication 152 (FIPS PUB 152) 26 September 1988.

## 2. FORMULAS

Mathematical notation even though it does not contain an equals sign is tagged in much the same way as though it were an equation or formula. From now on, the term 'formula' will be used to refer to formulas, equations, and mathematical notations.

At the outset it should be pointed out that 'a = b', '(a - b)', and other simple mathematical notation used in the examples through-out this guide are used solely for the sake of simplicity in illustrating tag use. When such mathematical notation can be entered directly into the text from the keyboard, clearly there is no need to tag it in terms of 'mathpac'.

All equations, formulas, and mathematical notations must be tagged as either an inline formula or a display formula as described in the following paragraphs.

## 2.1 INLINE FORMULAS

An inline formula is one that occurs in text. An example of an inline formula is:

'... when a = b, the fraction is not defined.'

Another example of an inline formula is:

'The value of the denominator (a - b) is significant.'

Inline formulas are tagged in terms of 'f'. These inline formulas are tagged as follows:

'... when <f>a = b</f>, the fraction is not defined.'

and

'The value of the denominator <f>(a - b)</f> is significant.'

## 2.2 DISPLAY FORMULAS

A display formula is one that is that is offset from the text. An example of a display formula is:

'... the sum

c = a + b

is significant.'

Another example of a display formula is:

'The difference

a - b

is negligible.'

Display formulas are tagged in terms of 'df'. These display formulas are tagged as follows:

'... the sum <df>c = a + b</df> is significant.'

and

'The difference <df>a - b</df> is negligible.'

A group of display formulas is tagged in terms of 'dfg'. Consider the following example:

'The sum and difference

(a + b)

(a - b)

are significant and negligible respectively.'

This example is tagged as:

'The sum and difference <dfg><df>(a + b)</df><df> (a - b)</df></dfg> are significant and negligible respectively.'

Note that the markup for display formulas can appear on the same line in the text since the FOSI will assure that the display formula is offset in the formatted text.

## 3. TAGGING VARIOUS FORMULAS

The preceding material gives a rough idea of what should be tagged, and what the two types of formulas are. We shall next consider the tagging of various mathematical constructs. The following examples for the various tagging procedures are admittedly very simple so as to illustrate the point in question. More complicated examples will be provided in Section 4 for reference.

The markup of certain 'mathpac' tags may contain certain general tags, such as the 'text' and 'operator' tags. The 'text' tags are 'italic', 'roman', and 'ov'. The first two 'text' tags refer to the respective type fonts. The 'ov' tag is used to provide 'embellishments' (such as '^', etc.) over tagged material. The 'operator' tags are: 'mark', 'markref', 'break', 'sup', 'sub', 'sum', 'integral', 'product', 'plex', 'frac', 'diff', 'sqrt', 'root', 'square', 'power', 'pile', 'matrix', 'fence', 'middle', 'tensor', 'mfn', 'box', and 'vec'. The use of the 'text' and 'operator' tags will be explained in the next section.

## 3.1 NOTATION

The following section describes the tagging of various mathematical notations ranging from the simple (subscripts, powers, roots, etc.) to the complex (derivatives, tensors, matrices, etc.).

### 3.1.1 SUPERSCRIPTS AND SUBSCRIPTS

The inline formula '... if we set $A_{ij}^{2}$=1 above ... ' is tagged as:

```
'... if we set <f>A<sup>2</sup><sub>
ij</sub> = 1</f> above ... '
```

This example introduces the use of the 'sup' and 'sub' tags for superscripts and subscripts respectively. Both 'sub' and 'sup' markups may contain 'text' and 'operator' tags. However, not all tagged quantities may contain 'sup' or 'sub' tags. As noted later, a vector (paragraph 3.1.9) may not have a superscript or a subscript nor may a function specified by an 'fname' tag in 'mfn' markup (paragraph 3.1.13).

### 3.1.2 FRACTIONS

The fraction $\frac{a}{b}$ is tagged as:

```
'<f><frac>a<over>b</frac></f>'.
```

The numerator appears between the 'frac' and 'over' tags, and the denominator between the 'over' and '</frac>' tags. The markups of the numerator and denominator may contain 'text' and 'operator' tags. The value of the 'align' attribute on the 'frac' tag specifies the placement of the fraction. If it is not specified, the fraction will be centered.

### 3.1.3 POWERS

The expression $(a+b)^3$ is tagged as:

'<f><power>3<of>(a + b)</power></f>'.

The exponent appears between the 'power' and the 'of' tags and the quantity being raised to that power between the 'of' and '</power>' tags. The markups of the exponent and the quantity being raised to that power may contain both 'text' and 'operator' tags.

The expression $(a+b)^2$ may also be tagged as:

'<f><square>(a + b)</square></f>'.

The quantity being squared appears between the 'square' and '</square>' tags. The markup of the quantity being squared may contain 'text' and 'operator' tags.

### 3.1.4 ROOTS

The cube root of a $\sqrt[3]{a}$ is tagged as:

'<f><root>3<of>a</root></f>'.

The root appears between the 'root' and 'of' tags and the radicand between the 'of' and '</root>' tags. The markups of the root and the radicand may contain 'text' and 'operator' tags.

The square root of a $\sqrt{a}$ is tagged as:

'<f><sqrt>a</sqrt></f>'.

The radicand appears between the 'sqrt' and '</sqrt>' tags. The markup of the radicand

may contain 'text' and 'operator' tags.

3.1.5 DERIVATIVES

The derivative $\frac{dy}{dx}$ is tagged as:

'<f><diff>y<by>x</diff></f>'.

The quantity being differentiated appears between the 'diff' and 'by' tags, and the variable of differentiation between the 'by' and '</diff>' tags. The markups of these two quantities may contain 'text' and 'operator' tags. The value of the 'type' attribute on the 'diff' tag specifies whether the derivative is ordinary or partial. If it is not specified, an ordinary derivative is assumed.

3.1.6 INTEGRALS

The integral $\int_a^b y dx$ is tagged as:

'<f><integral><from>a<to>b<of>
ydx</integral></f>'.

The lower limit of integration appears between the 'from' and 'to' tags, the upper limit of integration between the 'to' and 'of' tags, and the integrand between the 'of' and '</integral>' tags. The markups of these three quantities may contain 'text' and 'operator' tags.

3.1.7 SUMMATIONS

The summation $\sum_{i=1}^{N} a_i$ is tagged as:

'<f><sum><from>i=1<to>N<of>a
<sub>i</sub></sum></f>'.

The lower limit of summation appears between

the 'from' and 'to' tags, the upper limit of summation between the 'to' and 'of' tags, and the summand between the 'of' and '</sum>' tags. The markups of these three quantities may contain 'text' and 'operator' tags.

### 3.1.8 PRODUCTS

The treatment of products $\prod_{i=1}^{N} a_i$ is analogous to that of summations except that the 'product' tag is used. It would be tagged as:

'<f><product><from>i=1<to>N<of>a<sub>i</sub></sub></product></f>'.

### 3.1.9 VECTORS

The vector V is tagged as '<f><vec>V</vec></f>'. Vectors may not have superscripts or subscripts.

### 3.1.10 GENERAL OPERATORS

A general operator is tagged in terms of a 'plex' tag. For example, the 'union' set operation on a collection of N sets $S_i$, $\bigcup_{i=1}^{N} S_i$, is tagged as:

'<f><plex>∪<from>i=1<to>N<of>S<sub>i</sub></sub></plex></f>'

where the referencing of the 'ISOtech' entity 'cup' will provide the union operator character.

### 3.1.11 MATRICES

Matrices are tagged in terms of columns which are similar to piles (see Section 3.2.5). The matrix

$$\begin{vmatrix} 2 & 4 \\ 3 & 5 \end{vmatrix}$$

is tagged as '<df><matrix><col>2<above>3 </col><col>4<above>5</col></matrix></df>'. The markups of elements of a matrix may contain 'text' or 'operator' tags.

### 3.1.12 TENSORS

The markup of tensor notation is a bit complicated. The 'tensor' tag has two attributes. The first attribute 'posf' specifies whether the first character of the tensor's suffix is a superscript ('sup') or subscript ('sub'); its default value is 'sup'. The value of the second attribute 'suffix' consists of the characters of the suffix in the order they occur in the suffix separated or not separated by a space according to the following rule: if a character in the suffix changes from superscript to subscript or vice versa, then that character is separated from the preceding character by a space at that point in the value of the 'suffix' attribute.

Consider the following examples. The

tensor $A^{i\,k}_{j}$ is tagged as:

'<f><tensor suffix="i j k">A</tensor> </f>'.

Some explanation of this example will clarify the above rule for the 'suffix' attribute.

The first character of the suffix $^{i\,k}_{j}$ is 'i' which is clearly a superscript. Since the default value of the 'posf' attribute is 'sup', there is no need to specify the 'posf' attribute on the 'tensor' tag. Also the first character of the value of the 'suffix' attribute will be 'i'. However, the second character of the suffix is 'j' which is a subscript. This means that there must be a space between the 'i' and 'j' characters in the value of the 'suffix' attribute. Now the

third and last character of the suffix is 'k' which is a superscript. This means there must be a space between 'j' and 'k' characters in the value of the 'suffix' attribute. Accordingly, the value of the 'suffix' attribute on the 'tensor' tag is 'i j k'.

For a second example, consider the tensor $B_{mno}$ . It is tagged as '<f><tensor posf="sub" suffix="mno">B</tensor></f>.' In this case since all the characters of the suffix $_{mno}$ are subscripts, there are no intervening spaces in the 'mno' value of the 'suffix' attribute and the value of the 'posf' attribute must be specified as 'sub' on the 'tensor' tag.

### 3.1.13 FUNCTIONAL NOTATION

The 'mfn' function allows functions to be tagged in two different ways described below. For more details, the reader is referred to the 'mathpac' data dictionary in the appendix. It is assumed that the FOSI will provide appropriate formatting.

One way to tag a function is the following. If the name of the function is one of the following: 'and', 'antilog', 'arc', 'arccos', 'arcsin', 'arctan', 'arg', 'colog', 'cos', 'cosh', 'cot', 'coth', 'csc', 'ctn', 'deg', 'det', 'dim', 'exp', 'for', 'gcd', 'glb', 'hom', 'if', 'im', 'ker', 'lg', 'lim', 'ln', 'log', 'lub', 'max', 'min', 'mod', 're', 'sec', 'sin', 'sinh', 'tan', and 'tanh', then the name of the function is specified as the value of the 'type' attribute on the 'mfn' tag. For example, the functions 'cos (1 + x)' and 'max (x, y, z)' can be tagged as '<f><mfn type="cos">1 + x</mfn></f>' and '<f><mfn type="max">x, y, z</mfn></f>' respectively.

Another way to tag a function is the following. The name of the function is specified by 'fname' and 'of' tags. For example, the functions 'temp (T)' and 'plot

(x, y, z, t)' are tagged as '<f><mfn><fname>temp<of>T</mfn></f>' and '<f><mfn><fname>plot<of>x, y, z, t</mfn></f>' respectively.

## 3.2 PRESENTATION AND PLACEMENT

The following section describes the tagging of various mathematical notations with regard to their formatting (font, character embellishment, arrangement, placement, etc.).

### 3.2.1 FONTS

The 'roman' and 'italic' tags are used to provide their respective type font for those parts of a formula enclosed in these tags. Only alphanumeric text can be enclosed in these tags. The markups '<f><roman>A</roman></f>' and '<f><italic>A</italic></f>' will cause the mathematical notation 'A' to be printed in roman or italic font respectively.

### 3.2.2 CHARACTER EMBELLISHMENT

The 'ov' tag can be used to select twenty-one types of embellishment ('dot' through 'bar' - see the Appendix) to tagged quantities as specified by the 'type' attribute on the 'ov' tag with respect to the three values ('above', 'below', or 'mid') of the 'pos' (position) attribute and the six values ('single', 'double', 'triple', 'dash', 'dots', or 'bold') of the 'style' attribute on the 'ov' tag.

Consider the following markups. The markup '<f><ov>A</ov></f>' will result in the ' $\underline{A}$ ' symbol being formatted since the default values of the 'type', 'pos', and 'style' attributes are 'bar', 'above', and 'single' respectively. The markup '<f><ov pos="below">A</ov></f>' will result in the ' $\underline{A}$ ' symbol being formatted. The markup '<f><ov type="tilde">A</ov></f>' will result in ' $\tilde{A}$ ' being formatted. The markup '<f><ov type="acute"style="double">A</ov>

</f>' will result in the ' $A''$ ' symbol being formatted.

### 3.2.3 BREAKING FORMULAS

The 'break' tag is used to divide formulas. For example, the display formula

$$x + y + z = 25 + X + Y + Z$$

is tagged as '<df>x + y + z = 25 + <break>X + Y + Z</df>'.

### 3.2.4 ALIGNING FORMULAS

The 'mark' tag and the 'markref' tags are used to align formulas. For example, the

$$2X + 43Y = 142$$
$$113X + 2y = 26$$

system of simultaneous linear equations is tagged as:

```
'<dfg>
<df>2X + 43Y<mark id="here">= 142</df>
<df>113X + 2Y <markref refid="here">=
26</df>
</dfg>'
```

The same attribute value (which must belong with a letter) must be specified for the 'id' attribute on the 'mark' tag and the 'refid' attribute on the 'markref' tag.

### 3.2.5 COLUMNS ('PILES')

The column

a

b

c

is tagged as '<f><pile>a<above>b<above>c

</pile></f>'. The 'align' attribute of the 'pile' tag specifies the alignment of the pile's elements. These elements may be right or left justified or centered (default).

### 3.2.6 FENCES

Let us consider two examples to illustrate the 'fence' concept. First suppose the

fraction $\frac{3}{4}$ is to be enclosed in braces

('curly' bracket). The tagging for this would be '<f><fence type="brace"><frac>3<over>4 </frac></fence></f>'. An interval of the form ( ... ] is called a half open interval. The half open interval whose left (open) end is

$\frac{3}{4}$ and whose right (closed) end is $\frac{5}{6}$

is tagged as:

```
'<f><fence open="(" close="]">
<frac>3<over>4</frac>,<frac>5
<over>6</frac></fence></f>'.
```

In general the character used to 'fence off' (enclose) a quantity is specified by either the 'paren' (default), 'bracket', 'angbrack', 'brace', 'bar' or 'none' value of the 'type' attribute on the 'fence' tag. This character can be further specified by the 'single' (default), 'double', 'triple', 'dash', 'dots', or 'bold' value of the style attribute on the 'fence' tag. If different characters are used to 'fence off' a quantity, the leftmost enclosing character is specified by the value of the 'open' attribute on the 'fence' tag and the rightmost enclosing character by the value of the 'close' attribute on the 'fence' tag. For example of the use of different characters in 'fencing off' a quantity, see the above half open interval markup.

A 'post' (a vertical bar used to indicate choice) is entered using the 'middle' tag and its 'style' attribute similar to 'fence' above.

### 3.2.7 BOXES

The markup '<f><box>a + b</box></f>' will result in 'a + b' being enclosed in a box.

## 4. DETAILED EXAMPLES

Both of the following markups and their formatted hard copy are provided in section 8.6 'Complex examples' of TR 9573. The first example is that of a continued fraction. (The ' ... ' indicates that the value of the fraction is the limit of an infinite sequence of converging fractions.) This example is tagged as follows:

```
'<df>a<sub>0</sub>+<frac>b<sub>1</sub><over>a<sub>1
</sub>+<frac>b<sub>2</sub><over>a<sub>2</sub>+<frac>b
<sub>3</sub><over>a<sub>3</sub>+<frac>b<sub>4</sub>
<over>a<sub>4</sub>+...</frac></frac></frac></frac>
</df>'.
```

The second example is that of a product of an exponential and two fractions, one of which contains a partial derivative and the other an exponential. It is tagged as follows:

```
'<df>c<sub>&amp;mu;</sub>(t)=<power>-<frac>i<over>h</frac>E
<sub>&amp;mu;</sub>t<of>e</power>a<frac><fence><diff type=
"partial">V<by>a</diff></fence><sub>&amp;mu;0</sub><over>E
<sub>&amp;mu;</sub>-E<sub>0</sub></frac><frac><power>
<frac>i<over>h</frac>(E<sub>&amp;mu;</sub>-E<sub>0
</sub>)t<of>e</power>-1<over><frac>i<over>h</frac>
(E<sub>&amp;mu;</sub>-E<sub>0</sub>)</frac></df>'.
```

## 5. PROBLEMS

Two notable problems arise when using the 'mathpac' tags. The first results from the 'mixed content' in the content models of certain widely used 'mathpac' elements. The second is the inability of 'mathpac' tags to mark up certain commonly occurring mathematical notations.

## 5.1 'MIXED CONTENT' PROBLEMS

There are several problems with 'mathpac'. First three 'mathpac' elements 'operator', 'mfn', and 'middle' have 'mixed content'. This means that carriage return and spaces between tags may be considered as 'data' under certain circumstances, instead of being ignored by a parser conforming to ISO 8879. This means that apparently acceptable markups of these elements may not parse.

## 5.2 MATHPAC TAGGING PROBLEMS

Unfortunately the 'mathpac' tagset is incomplete and provides no way to tag some commonly used mathematical notations. The following list is based on the present limited investigation of the 'mathpac' tags. There may well be other such deficiencies.

a. The 'mathpac' tagset does not support vectors which have subscripts or superscripts. There seems to be no way to get around this.

b. Function names tagged with the 'fname' tag in 'mfn' markups may not have subscripts or superscripts.

c. There is no way of tagging second and higher order derivatives, other than tagging them as fractions. For example, the second derivative of y with respect to x must be tagged as

```
'<df><frac>d<sup>2</sup>y<over>dx
<sup>2</sup></frac></df>'.
```

Moreover, the tagging of such higher order partial derivatives requires the use of the ISOtech 'part' entity to provide the partial differential character.

APPENDIX A. MATHPAC DATA DICTIONARY

TABLE I. MATHPAC TAG SET DESCRIPTIONS

| Element/ Attribute | Full name | Description |
|---|---|---|
| <above> | Above | Identifies the next element in a vertical list. |
| <above1> | Above1 | Identifies the top element in a vertical list. Usually, the start and end tag may be omitted. This is followed by one or more above elements, where the start tag is required. |
| <box style =x> | Box | Identifies parts of a formula to be placed inside a box.<br><br>Optional Attribute(s)<br>STYLE: Style of the box's rules declared in the "f.style" entity.<br>Declared Value = single, double, triple, dash, dots, or bold.<br>Default = "single" |
| <break type =x> | Formula Breakpoint | Identifies a division point (for horizontal alignment break) within a formula.<br><br>Optional Attribute(s)<br>TYPE: Specifies whether formula breakpoint is required or not.<br>Declared Value = required or optional.<br>Default = "required" |
| <by> | Derivative Denominator | Identifies the denominator of a derivative. |

| | | |
|---|---|---|
| <col<br>align =x> | Matrix Column | Identifies a column of a matrix.<br><br>Optional Attribute(s)<br>ALIGN: Horizontal alignment of the characters within a matrix as defined in the "f.align" entity.<br>Declared Value = center, left, or right.<br>Default = "center" |
| <degree> | Degree | Identifies a degree. The sum of the exponents of the variable factors of the term of highest degree in a polynomial. |
| <df<br>id =x<br>align =x<br>num =x> | Display Formula | Identifies a display formula.<br><br>Optional Attribute(s)<br>ID: Unique identifier of the display formula.<br>Declared Value = ID<br>Default = IMPLIED (NULL)<br>ALIGN: Horizontal alignment of the display formula as declared in the "f.align" entity.<br>Declared Value = center, left, or right.<br>Default = "left"<br>NUM: Specifies an explicit formula number; if omitted, sequential numbering of the formulae would normally be performed by the text formatter.<br>Declared Value = CDATA<br>Default = IMPLIED (NULL) |
| <dfg<br>id =x<br>align =x<br>num =x> | Display Formula Group | Identifies a formula group whose content is display formulae.<br><br>Optional Attribute(s)<br>ID: Unique identifier of the |

| | | |
|---|---|---|
| | | display formula.<br>Declared Value = ID<br>Default = IMPLIED (NULL)<br><br>ALIGN: Horizontal alignment of the display formula as declared in the "f.align" entity.<br>Declared Value = center, left, or right,<br>Default = "left"<br>NUM: Specifies an explicit formula group number; if omitted, sequential numbering of the formulae would normally be performed by the text formatter.<br>Declared Value = CDATA<br>Default = IMPLIED (NULL) |
| <dfref<br>page =x<br>refid =x> | Formula Reference | Identifies a reference to a formula or formula group.<br><br>Required Attribute(s)<br>REFID: Reference to the unique identifier of a <df>.<br>Declared Value = IDREF<br><br>Optional Attribute(s)<br>PAGE: Page number is added (default) to the reference of the unique identifier of a <df>.<br>Declared Value = yes or no.<br>Default = "yes" |
| <diff<br>type =x> | Differential | Differential tag that identifies a derivative.<br><br>Optional Attribute(s)<br>TYPE: Type of differential as declared in the "f.diff" entity. |

| | | |
|---|---|---|
| | | Declared Value = normal or partial.<br>Default = "normal" |
| <diffof> | Derivative Numerator | Identifies the numerator of the derivative. |
| <f> | Inline Formula | Identifies an inline formula. |
| <fence<br>type =x<br>open =x<br>close =x<br>style =x> | Fence | Identifies a fence (e.g., open bracket).<br><br>Optional Attribute(s)<br>TYPE: Identifies the character to be used for the fence.<br>Declared Value = paren, bracket, angbrack, brace, bar, or none.<br>Default = "paren"<br>OPEN: Unpaired fences, also single occurrences, are entered using the <fence> element and specifying the open and close attributes. The value of these attributes is the fence character to be used.<br>Declared Value = CDATA<br>Default = IMPLIED (NULL)<br>CLOSE: Unpaired fences, also single occurrences, are entered using the <fence> element and specifying the open and close attributes. The value of these attributes is the fence character to be used.<br>Declared Value = CDATA<br>Default = IMPLIED (NULL)<br>STYLE: Defines the style of the fence whose values are declared in the "f.style" entity.<br>Declared Value = single, double, triple, dash, dots, or |

| | | |
|---|---|---|
| | | bold.<br>Default = "single" |
| <fname> | Function Name | Identifies a function name associated with the "<mfn>" tag. |
| <frac<br>align = x > | Fraction | Identifies a fraction.<br><br>Optional Attribute(s)<br>ALIGN: Horizontal alignment of the fraction as declared in the "f.align" entity.<br>Declared Value = center, left, or right.<br>Default = "center" |
| <from> | From | Identifies mathematical lower limit. |
| <integral> | Integral | Identifies an integral (special case of the general plex). |
| <italic> | Italic | By convention, most characters except numeral digits and syntax elements are set in italics in mathematical formulae. Exceptions to this convention include names of functions. |
| <mark<br>id =x> | Mark | Identifies a division point (for horizontal alignment break) within a formula.<br><br>Required Attribute(s)<br>ID: Used when referring to a mark and must be a unique reference in the document.<br>Declared Value = ID |

| | | |
|---|---|---|
| <markref refid =x> | Mark Reference | Identifies a division point (for horizontal alignment break) within a formula. It causes the system to align the formula horizontally on that mark.<br><br>Required Attribute(s)<br>REFID: Value is a valid identifier to which the markref refers.<br>Declared Value = IDREF |
| <matrix> | Matrix | Identifies a matrix. They are different from piles in that row elements of the matrix will always align horizontally rather than just occupy the required vertical space in the column. |
| <mfn type =x> | Mathematical Function | Identifies a mathematical function and its argument. By convention, most characters except numeral digits and syntax elements are set in italics in mathematical formulae. Exceptions to this convention include names of functions.<br><br>Optional Attribute(s)<br>TYPE: Type of mathematical function whose values are declared in the "f.func" entity.<br>Declared Value = and, antilog, arc, arccos, arcsin, arctan, arg, colog, cos, cosh, cot, coth, csc, ctn, deg, det, dim, exp, for, gcd, glb, hom, if, im, key, lg, lim, ln, log, lub, |

| | | |
|---|---|---|
| | | max, min, mod, re, sec, sin, sinh, tan, or tanh.<br>Default = IMPLIED (NULL) |
| <middle style =x> | Middle | Identifies a post.<br><br>Optional Attribute(s)<br>STYLE: Style of the post as declared in the "f.style" entity.<br>Declared Value = single, double, triple, dash, dots, or bold.<br>Default = "single" |
| <numer> | Numerator | Identifies the numerator of the fraction. |
| <of> | Limit Of Plex | Identifies limit for a plex. Used with <from> and <to> to define mathematical upper and lower limits. |
| <operator> | Operator | Identifies an mathematical operator. |
| <over> | Denominator | Identifies the denominator of a fraction. |
| <ov type =x pos =x style =x> | "Over" Embellishments | Identifies parts of a formula over which special accents or diacritical marks are to be placed. (Over-Character Tag)<br><br>Optional Attribute(s)<br>TYPE: Defines type of mark whose values are declared in the "f.ov" entity.<br>Declared Value = dot, dotdot, dot3, dot4, tie, tiebrace, hat, caron, acute, grave, cedil, ring, macron, ogonek, dblac, breve, tilde, vec, |

| | | |
|---|---|---|
| | | rvec, dyad, or bar.<br>Default = "bar"<br>POS: Defines position of the mark.<br>Declared Value = above, below, or mid.<br>Default = "above"<br>STYLE: Defines style of the mark as declared in the "f.style" entity.<br>Declared Value = single, double, triple, dash, dots, or bold.<br>Default = "single" |
| <pile align =x> | Pile | Identifies a vertical grouping of segments of a formula. Commonly used for arranging equations in columns (i.e., fibonacci sequence).<br><br>Optional Attribute(s)<br>ALIGN: Horizontal alignment of the pile as declared in the "f.align" entity.<br>Declared Value = center, left, or right.<br>Default = "center" |
| <plex> | Plex | Identifies a general "limits" operator. |
| <power> | Power | Identifies a power. The number of times as indicated by an exponent that a number occurs as a factor in a product. |

| | | |
|---|---|---|
| <product> | Product | Identifies a product (special case of the general plex). |
| <roman> | Roman | Identifies the rare case where parts of a formula should be set in a roman font, contrary common practice and not being a function name. |
| <root> | Root | Identifies a root. (The nth root of the number "X", is defined to be a number "Y" such that "Y" to the "N" equals "X"). |
| <sqrt> | Square Root | Identifies a square root. (The square root of the number "X", is defined to be a number "Y" such that "Y" multiplied by itself equals "X".) |
| <square> | Square | Identifies a square. The product of a number multiplied by tself. |
| <sub pos =x> | Subscript | Indicates a subscript. For use only within mathematical formulae. NOTE: Within text use <subscrpt> instead.<br><br>Optional Attribute(s)<br>POS: Position of subscript.<br>Declared Value = pre, mid, or post.<br>Default = "post" |
| <sum> | Sum | Identifies a sum (special case of the general plex). |

| | | |
|---|---|---|
| <sup pos =x> | Superscript | Indicates a superscript. For use only within mathematical formulae. NOTE: Within text use <supscrpt> instead.<br><br>Optional Attribute(s)<br>POS: Position of superscript.<br>Declared Value = pre, mid, or post.<br>Default = "post" |
| <tensor posf =x suffix =x> | Tensor | Identifies a tensor. A generalized vector with more than three components each of which is a function of the coordinates of an arbitrary point in space of an appropriate number of dimensions.<br><br>Required Attribute(s)<br>SUFFIX: The suffix attribute takes the suffixes of the tensor; a space indicating a switch from superscript to subscript or subscript to superscript.<br>Declared Value = CDATA<br><br>Optional Attribute(s)<br>POSF: The value of the position of the first suffix attribute.<br>Declared Value = sub (subscript) or sup (superscript).<br>Default = "sup" |
| <to> | To | Identifies mathematical upper limit. |

| | | |
|---|---|---|
| <vec> | Vector | Identifies a vector in a formula. By convention these are set in bold roman or medium italic with an arrow above. |

TABLE II. MATHPAC ATTRIBUTE SET DESCRIPTIONS

| Attribute | Full Name | Description |
|---|---|---|
| %f.align; | Mathematical Formula Alignment | Defines the mathematical alignment position within a formula.<br><br>Optional Attribute(s)<br>ALIGN: Defines the values declared in the "f.align" entity.<br>Declared Value = center, left, or right. |
| %f.diff; | Differential | Differential tag that identifies a derivative.<br><br>Optional Attribute(s)<br>TYPE: Type of differential as declared in the "f.diff" entity.<br>Declared Value = normal or partial |
| %f.func; | Mathematical Function | Identifies a mathematical function and its argument. By convention, most characters except numeral digits and syntax elements are set in italics in mathematical formulae. Exceptions to this convention include names of functions.<br><br>Optional Attribute(s) |

| | | |
|---|---|---|
| | | TYPE: Defines the type of mathematical function whose values are declared in the "f.func" entity.<br>Declared Value = and, antilog, arc, arccos, arcsin, arctan, arg, colog, cos, cosh, cot, coth, csc, ctn, deg, det, dim, exp, for, gcd, glb, hom, if, im, key, lg, lim, ln, log, lub, max, min, mod, re, sec, sin, sinh, tan, or tanh. |
| %f.oper; | Operator | Identifies an operator within a formula. Defined by the values declared in the "f.oper" entity.<br>Valid operators are: = mark, markref, break, sup, sub, sum, integral, product, plex, frac, diff, sqrt, root, square, power, pile, matrix, middle, tensor, mfn, box, or vec. |
| %f.ov; | "Over" Embellishments | Identifies parts of a formula over which special accents or diacritical marks are to be placed. (Over-Character Tag)<br><br>Optional Attribute(s)<br>TYPE: Defines the type of mark whose values are declared in the "f.ov" entity.<br>Declared Value = dot, dotdot, dot3, dot4, tie, tiebrace, hat, haczeck, acute, grave, cedil, ring, macron, ogonek, dblac, breve, tilde, vec, rvec, dyad, or bar. |
| %f.pos; | Position of Elements | Identifies position of elements within a formula containing superscripts and subscripts. |

| | | |
|---|---|---|
| | | Optional Attribute(s)<br>TYPE: Defines the position of elements whose values are declared in the "f.pos" entity.<br>Declared Value = pre, mid, or post. |
| %f.style; | Style | Defines a mathematical style to be used with a formula.<br><br>Optional Attribute(s)<br>STYLE: Style of rules declared in the "f.style" entity.<br>Declared Value = single, double, triple, dash, dots, or bold. |
| %f.text; | Type of Text | Identifies the type of text to be used. Defines the type of text whose values are declared in the "f.text" entity. Valid values are: = #PCDATA, roman, italic, or ov. |
| %f.type; | Type of Fence | Identifies the character that will be used as a fence (e.g., open bracket).<br><br>Optional Attribute(s)<br>TYPE: Identifies the character to be used for the fence.<br>Declared Value = paren, bracket, angbrack, brace, bar, or none. |