

NAME

eqn, neqn, checkeq — typeset mathematical text

SYNOPSIS

```
eqn [ -dxy ] [ -pn ] [ -sn ] [ -fn ] [ file ] ...
neqn [ -dxy ] [ -pn ] [ -sn ] [ -fn ] [ file ] ...
checkeq [ file ] ...
```

DESCRIPTION

Eqn is a *troff*(1) preprocessor for typesetting mathematical text on a Wang-Graphic Systems, Inc. phototypesetter, while *neqn* is used for the same purpose with *nroff*(1) on typewriter-like terminals. Usage is almost always:

```
eqn file ... | troff
neqn file ... | nroff
```

If no files are specified, these programs read from the standard input. A line beginning with *.EQ* marks the start of an equation; the end of an equation is marked by a line beginning with *.EN*. Neither of these lines is altered, so they may be defined in macro packages to get centering, numbering, etc. It is also possible to designate two characters as *delimiters*; subsequent text between delimiters is then treated as *eqn* input. Delimiters may be set to characters *x* and *y* with the command-line argument *-dxy* or (more commonly) with *delim xy* between *.EQ* and *.EN*. The left and right delimiters may be the same character; the dollar sign is often used as such a delimiter. Delimiters are turned off by *delim off*. All text that is neither between delimiters nor between *.EQ* and *.EN* is passed through untouched.

The program *checkeq* reports missing or unbalanced delimiters and *.EQ/.EN* pairs.

Tokens within *eqn* are separated by spaces, tabs, new-lines, braces, double quotes, tildes, or circumflexes. Braces *{}* are used for grouping; generally speaking, anywhere a single character such as *x* could appear, a complicated construction enclosed in braces may be used instead. Tilde *~* represents a full space in the output, circumflex *^* half as much.

Subscripts and superscripts are produced with the keywords *sub* and *sup*. Thus *x sub j* makes x_j , *a sub k sup 2* produces a_k^2 , and *e sup {x sup 2 + y sup 2}* gives $e^{x^2+y^2}$.

Fractions are made with *over*: *a over b* yields $\frac{a}{b}$.

sqrt makes square roots: *1 over sqrt {ax sup 2+bx+c}* results in $\frac{1}{\sqrt{ax^2+bx+c}}$.

The keywords *from* and *to* introduce lower and upper limits on arbitrary things: $\lim_{n \rightarrow \infty} \sum_0^n x_i$ is made with *lim from {n-> inf} sum from 0 to n x sub i*.

Left and right brackets, braces, etc., of the right height are made with *left* and *right*:

left [x sup 2 + y sup 2 over alpha right] ~ = ~ 1 produces $\left[x^2 + \frac{y^2}{\alpha} \right] = 1$. The *right* clause is optional. Legal characters after *left* and *right* are braces, brackets, bars, *c* and *f* for ceiling and floor, and *"* for nothing at all (useful for a right-side-only bracket).

Vertical piles of things are made with *pile*, *lpile*, *cpile*, and *rpile*: *pile {a above b above c}* produces $\begin{matrix} a \\ b \\ c \end{matrix}$. There can be an arbitrary number of elements in a pile; *lpile* left-justifies, *pile* and *cpile* center (but with different vertical spacing), and *rpile* right justifies.

Matrices are made with **matrix**: `matrix { lcol { x sub i above y sub 2 } ccol { 1 above 2 } }`
 produces
$$\begin{matrix} x_i & 1 \\ y_2 & 2 \end{matrix}$$
. In addition, there is **rcol** for a right-justified column.

Diacritical marks are made with **dot**, **dotdot**, **hat**, **tilde**, **bar**, **vec**, **dyad**, and **under**: x **dot** = $f(t)$
 \bar{bar} is $\dot{x} = \dot{f}(t)$, y **dotdot** \bar{bar} = \ddot{y} , n **under** is $\bar{y} = \underline{n}$, and x **vec** = \vec{x} , y **dyad** is $\bar{x} = \bar{y}$.

Point sizes and fonts can be changed with **size** n or **size** $\pm n$, **roman**, **italic**, **bold**, and **font** n .
 Point sizes and fonts can be changed globally in a document by **gsize** n and **gfont** n , or by the
 command-line arguments **-sn** and **-fn**.

Normally, subscripts and superscripts are reduced by 3 points from the previous size; this may
 be changed by the command-line argument **-pn**.

Successive display arguments can be lined up. Place **mark** before the desired lineup point in
 the first equation; place **lineup** at the place that is to line up vertically in subsequent equations.

Shorthands may be defined or existing keywords redefined with **define**:

`define thing % replacement %`

defines a new token called *thing* that will be replaced by *replacement* whenever it appears
 thereafter. The % may be any character that does not occur in *replacement*.

Keywords such as **sum** (Σ), **int** (\int), **inf** (∞), and shorthands such as **>=** (\geq), **!=** (\neq), and
-> (\rightarrow) are recognized. Greek letters are spelled out in the desired case, as in **alpha** or
GAMMA. Mathematical words such as **sin**, **cos**, and **log** are made Roman automatically.
Troff(1) four-character escapes such as **\(dd** (\ddagger) and **\(bs** (\textcircled{B}) may be used anywhere. Strings
 enclosed in double quotes "..." are passed through untouched; this permits keywords to be
 entered as text, and can be used to communicate with **troff(1)** when all else fails. Full details
 are given in the manual cited below.

SEE ALSO

Typesetting Mathematics — User's Guide by B. W. Kernighan and L. L. Cherry
mm(1), **mmt(1)**, **tbl(1)**, **troff(1)**, **eqnchar(7)**, **mm(7)**, **mv(7)**.

BUGS

To embolden digits, parens, etc., it is necessary to quote them, as in **bold "12.3"**.