NAME

eqn, neqn, checkeq – typeset mathematics test

SYNOPSIS

eqn [–**d***xy*] [–**f***n*] [–**p***n*] [–**s***n*] [*filename*] . . .

neqn [*filename*] . . .

checkeq [*filename*] . . .

AVAILABILITY

SUNWdoc

DESCRIPTION

eqn and **neqn** are language processors to assist in describing equations. **eqn** is a preprocessor for **troff**(1) and is intended for devices that can print **troff**'s output. **neqn** is a preprocessor for **nroff**(1) and is intended for use with terminals. Usage is almost always:

example% eqn *filename* **. . . | troff**
example% neqn *filename* **. . . | nroff**

If no *filename*s are specified, **eqn** and **neqn** 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 set two characters as ''delimiters''; subsequent text between delimiters is also treated as **eqn** input.

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

OPTIONS

–**d***xy* Set equation delimiters set to characters *x* and *y* with the command-line argument. The more common way to do this is with **delim** *xy* between **.EQ** and **.EN**. The left and right delimiters may be identical. Delimiters are turned off by **delim off** appearing in the text. All text that is neither between delimiters nor between **.EQ** and **.EN** is passed through untouched.

–**f***n* Change font to *n* globally in the document. The font can also be changed globally in the body of the document by using the **gfont** *n* directive, where *n* is the font specification.

–**p***n* Reduce subscripts and superscripts by *n* point sizes from the previous size. In the absence of the –**p** option, subscripts and superscripts are reduced by 3 point sizes from the previous size.

–**s***n* Change point size to *n* globally in the document. The point size can also be changed globally in the body of the document by using the **gsize** *n* directive, where *n* is the point size.

EQN LANGUAGE

NOTE: The nroff version of this description depicts the output of **neqn** to the terminal screen exactly as **neqn** is able to display it. To see an accurate depiction of the output the printed version of this page should be viewed.

Tokens within **eqn** are separated by braces, double quotes, tildes, circumflexes, SPACE, TAB, or NEWLINE characters. Braces { } are used for grouping; generally speaking, anywhere a single character like *x* could appear, a complicated construction enclosed in braces may be used instead. Tilde (˜) represents a full SPACE in the output, circumflex (ˆ)

half as much.

Subscripts and superscripts:
These are produced with the keywords **sub** and **sup**.

x sub i makes x_i

a sub i sup 2 produces a_i^2

e sup {x sup 2 + y sup 2} gives $e^{x^2+y^2}$

Fractions:
Fractions are made with **over**.

a over b yields $\frac{a}{b}$

Square Roots:
These are made with **sqrt**

1 over sqrt {ax sup 2 +bx+c} results in

$$\frac{1}{\sqrt{ax^2+bx+c}}$$

Limits: The keywords **from** and **to** introduce lower and upper limits on arbitrary things:
lim from {n→ inf } sum from 0 to n x sub i makes

$$\lim_{n\to\infty}\sum_0^n x_i$$

Brackets and Braces:
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:
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, with different vertical spacing, and **rpile** right justifies.

Matrices:

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:

Diacritical marks are made with **dot**, **dotdot**, **hat**, **tilde**, **bar**, **vec**, **dyad**, and **under**.

x dot = f(t) bar is

$$\dot{x} = \overline{f(t)}$$

y dotdot bar ˜=˜ n under is

$$\bar{\ddot{y}} = \underline{n},$$

x vec ˜=˜ y dyad is

$$\vec{x} = \overleftrightarrow{y}.$$

Sizes and Fonts:

Sizes and font can be changed with **size** *n* or **size** ±*n*, **roman**, **italic**, **bold**, and **font** *n*. Size and fonts can be changed globally in a document by **gsize** *n* and **gfont** *n*, or by the command-line arguments –**s***n* and –**f***n*.

Successive display arguments:

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:

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

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

Keywords and Shorthands:

Keywords like **sum** (∑), **int** (∫), **inf** (∞), and shorthands like **>=** (≥), → (→), and **!=** (≠) are recognized.

Greek letters:

Greek letters are spelled out in the desired case, as in **alpha** or **GAMMA**.

Mathematical words:
Mathematical words like **sin**, **cos**, and **log** are made Roman automatically.

troff(1) four-character escapes like \(bu (•) can 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** when all else fails.

SEE ALSO

tbl(1), **troff**(1), **ms**(5)

BUGS

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