FEET, INCHES, & FRACTIONS (FIF)

Fun with architectural & structural engineering dimensions

FRED LUSK, P.E.

INTRODUCTION {1}

The purpose of this worksheet is to demonstrate various Mathcad techniques for dealing with length dimensions in feet-inches-fractions (FIF) format. Because the USA's ponderous conversion to the SI system is limited mostly to the federal government, most architectural, structural engineering, and plant piping dimensions in the United States are still provided in the "quaint" US Customary Units (USCU) system, with the added burden of using FIF values instead of decimal feet like we do for surveying and most non-structural/non-piping civil engineering.

Because of Mathcad's brilliant handling of units, it deals with FIF values as well or better than any other tool I know of, including FIF-capable calculators (which, sadly, are not RPN). Unfortunately, Mathcad does not have an elegant way to display results in FIF format. However, this can be done after a fashion, as this worksheet shows. Mathcad also cannot display numbers in degrees-minutes-seconds (DMS) and hours-minutes-seconds (HMS) formats, but that is a topic for another day.

Four data input methods are shown: (a) using Mathcad's FIF function on dimensions provided as strings or sets of three scalar values, (b) using the simple addition of feet, inches, and fractions, (c) using a four-element vector, and (d) using a table with multiple FIF dimensions. Two versions of a program are included to break down decimal lengths into feet, inches, and reduced fractions and then display the FIF components. Various examples are provided.

This worksheet is based ultimately on an FIF program I wrote in 1984 for the HP-41CX calculator (see http://www.hpmuseum.org/software/41/41feet.htm). The HP-41CX program performs FIF artithmetic using a two-register RPN stack, can use ANY counting number as the basis for fractions of an inch, and can even use trig functions, which is useful for FIF triangle solutions. I later upgraded this program by simplifying the algorithm, but there is one minor bug I should be chasing so I haven't published it yet. There is also an HP-42S version of the upgraded program that is also unpublished due to the same bug. Fixing these programs is on my never-ending to-do list.

In 2004, I created an Excel spreadsheet for adding and subtracting multiple FIF values and for calculating areas and volumes. I have included a copy of this spreadsheet (in template form) for your use. An architect friend, whom I supplied this to years ago, told me recently that he still uses it.

The calculator programs are the most clever (the fraction reduction subroutine is based on the **FR** program in the famous PPC ROM) and produce the best formatted output, but they are very slow compared to the virtually instantaneous results provided by the Excel spreadsheet and this Mathcad worksheet. Because Mathcad natively handles units (unlike Excel) , this worksheet contains the simplest algorithms of the three. However, Mathcad has the weakest formatting tools for FIF, so this is one of the few occasions where I think the Excel spreadsheet is more usable than I will ever be able to make this Mathcad worksheet. On the other hand, if you need FIF capabilities in a Mathcad worksheet, then there is no substitute for having it, which this worksheet provides.

DATA {2}

Fraction of an inch to round off to: $FR := 16$ $Fraction := \frac{1}{FR}\ in$

(Any counting number will do.)

GENERAL PROGRAMS & FUNCTIONS

Greatest Common Divisor: [1,2] {3}

(Used for fraction reduction in the FT2FIF and FT2FIFA function programs.)

Function program:

$$\mathrm{GCD}(n_1, n_2) := \left\| \begin{array}{l} \text{while } n_2 \neq 0 \\ \quad \left\| \begin{array}{l} t \leftarrow n_2 \\ n_2 \leftarrow \mathrm{mod}(n_1, n_2) \\ n_1 \leftarrow t \end{array} \right. \\ n_1 \end{array} \right.$$

Examples:

$\mathrm{GCD}(6, 16) = 2$ $\mathrm{GCD}(81, 1107) = 27$ $\mathrm{GCD}(17, 73) = 1$

Convert decimal feet to FIF vector: {4}

(Results are presented in a unitless 4-element vector: feet, inches, and rounded & reduced numerator, & denominator.)

Function program:

$$\mathrm{FT2FIF}(x) := \left\| \begin{array}{l} Feet \leftarrow \mathrm{Trunc}(x, 1\ ft) \\ FT \leftarrow \dfrac{Feet}{ft} \\ Inches \leftarrow \mathrm{Trunc}(x - Feet, 1\ in) \\ IN \leftarrow \dfrac{Inches}{in} \\ Numerator \leftarrow \mathrm{Round}(FR \cdot (x - Feet - Inches), 1\ in) \\ NUM \leftarrow \mathrm{round}\left(\dfrac{Numerator}{in}, 0\right) \\ n \leftarrow NUM \\ f \leftarrow FR \\ d \leftarrow \mathrm{GCD}(n, f) \\ FIF \leftarrow \begin{bmatrix} FT \\ IN \\ \dfrac{NUM}{d} \\ \dfrac{FR}{d} \end{bmatrix} \end{array} \right.$$

Examples:

$\mathrm{FT2FIF}(2.75\ ft) = \begin{bmatrix} 2 \\ 9 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} \text{"ft"} \\ \text{"in"} \\ \text{"num"} \\ \text{"den"} \end{bmatrix}$ $\mathrm{FT2FIF}(2.068\ ft) = \begin{bmatrix} 2 \\ 0 \\ 13 \\ 16 \end{bmatrix} \begin{bmatrix} \text{"ft"} \\ \text{"in"} \\ \text{"num"} \\ \text{"den"} \end{bmatrix}$

$\mathrm{FT2FIF}(63.25\ in) = \begin{bmatrix} 5 \\ 3 \\ 1 \\ 4 \end{bmatrix} \begin{bmatrix} \text{"ft"} \\ \text{"in"} \\ \text{"num"} \\ \text{"den"} \end{bmatrix}$ $\mathrm{FT2FIF}(3.750\ m) = \begin{bmatrix} 12 \\ 3 \\ 5 \\ 8 \end{bmatrix} \begin{bmatrix} \text{"ft"} \\ \text{"in"} \\ \text{"num"} \\ \text{"den"} \end{bmatrix}$

$\mathrm{FT2FIF}(6.333\ in) = \begin{bmatrix} 0 \\ 6 \\ 5 \\ 16 \end{bmatrix} \begin{bmatrix} \text{"ft"} \\ \text{"in"} \\ \text{"num"} \\ \text{"den"} \end{bmatrix}$ $\mathrm{FT2FIF}(0.167\ in) = \begin{bmatrix} 0 \\ 0 \\ 3 \\ 16 \end{bmatrix} \begin{bmatrix} \text{"ft"} \\ \text{"in"} \\ \text{"num"} \\ \text{"den"} \end{bmatrix}$

Convert decimal feet to annotated FIF matrix: {4}

(This is similar to the previous program function, but with "units" included in the results matrix.)

Function program:

$$\mathrm{FT2FIFA}(x) := \left\| \begin{array}{l} Feet \leftarrow \mathrm{Trunc}(x, 1\ ft) \\ FT \leftarrow \dfrac{Feet}{ft} \\ Inches \leftarrow \mathrm{Trunc}(x - Feet, 1\ in) \\ IN \leftarrow \dfrac{Inches}{in} \\ Numerator \leftarrow \mathrm{Round}(FR \cdot (x - Feet - Inches), 1\ in) \\ NUM \leftarrow \mathrm{round}\left(\dfrac{Numerator}{in}, 0\right) \\ n \leftarrow NUM \\ f \leftarrow FR \\ d \leftarrow \mathrm{GCD}(n, f) \\ FIF \leftarrow \begin{bmatrix} FT & \text{“ft”} \\ IN & \text{“in”} \\ \dfrac{NUM}{d} & \text{“num”} \\ \dfrac{FR}{d} & \text{“den”} \end{bmatrix} \end{array} \right.$$

Examples:

$$\mathrm{FT2FIFA}(2.75\ ft) = \begin{bmatrix} 2 & \text{“ft”} \\ 9 & \text{“in”} \\ 0 & \text{“num”} \\ 1 & \text{“den”} \end{bmatrix} \qquad \mathrm{FT2FIFA}(2.068\ ft) = \begin{bmatrix} 2 & \text{“ft”} \\ 0 & \text{“in”} \\ 13 & \text{“num”} \\ 16 & \text{“den”} \end{bmatrix}$$

$$\mathrm{FT2FIFA}(63.25\ in) = \begin{bmatrix} 5 & \text{“ft”} \\ 3 & \text{“in”} \\ 1 & \text{“num”} \\ 4 & \text{“den”} \end{bmatrix} \qquad \mathrm{FT2FIFA}(3.750\ m) = \begin{bmatrix} 12 & \text{“ft”} \\ 3 & \text{“in”} \\ 5 & \text{“num”} \\ 8 & \text{“den”} \end{bmatrix}$$

$$\mathrm{FT2FIFA}(6.3333\ in) = \begin{bmatrix} 0 & \text{“ft”} \\ 6 & \text{“in”} \\ 5 & \text{“num”} \\ 16 & \text{“den”} \end{bmatrix} \qquad \mathrm{FT2FIFA}(0.167\ in) = \begin{bmatrix} 0 & \text{“ft”} \\ 0 & \text{“in”} \\ 3 & \text{“num”} \\ 16 & \text{“den”} \end{bmatrix}$$

Convert unitless FIF vector to decimal feet: {5}

(The subscripts assume $a = ORIGIN$. Mathcad isn't fazed by the oddball dimensions in the second example.)

Function:

$$\mathrm{FIF2FT}(v) := \left(v_a + \frac{v_{a+1} + \dfrac{v_{a+2}}{v_{a+3}}}{12} \right) ft$$

Examples:

$$\mathrm{FIF2FT}\left(\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}\right) = 1.2292\ ft \qquad \mathrm{FIF2FT}\left(\begin{bmatrix} 10 \\ 15 \\ 30 \\ 20 \end{bmatrix}\right) = 11.3750\ ft$$

Convert FIF vector with units to decimal feet: {5}

(The subscripts assume $a = ORIGIN$. The *num* and *den* units are defined in the section on Global Definitions.)

Function:

$$\text{FIFU2FT}(v) := \left(v_a + v_{a+1} + \frac{v_{a+2}\ in}{v_{a+3}} \right) \qquad \text{FIFU2FT}\left(\begin{bmatrix} 1\ ft \\ 3\ in \\ 0\ num \\ 1\ den \end{bmatrix}\right) = 1.25\ ft$$

Examples:

$$\text{FIFU2FT}\left(\begin{bmatrix} 1\ ft \\ 2\ in \\ 3\ num \\ 4\ den \end{bmatrix}\right) = 1.2292\ ft \qquad \text{FIFU2FT}\left(\begin{bmatrix} 10\ ft \\ 15\ in \\ 30\ num \\ 20\ den \end{bmatrix}\right) = 11.3750\ ft$$

Convert annotated FIF matrix to decimal feet: {5}

(The subscripts assume $a = ORIGIN$.)

Function:

$$\text{FIFA2FT}(v) := \left(v_{a,a} + \frac{v_{a+1,a} + \frac{v_{a+2,a}}{v_{a+3,a}}}{12} \right) ft$$

Examples:

$$\text{FIFA2FT}\left(\begin{bmatrix} 1 & \text{“ft”} \\ 2 & \text{“in”} \\ 3 & \text{“num”} \\ 4 & \text{“den”} \end{bmatrix}\right) = 1.2292\ ft \qquad \text{FIFA2FT}\left(\begin{bmatrix} 10 & \text{“ft”} \\ 15 & \text{“in”} \\ 30 & \text{“num”} \\ 20 & \text{“den”} \end{bmatrix}\right) = 11.3750\ ft$$

Convert length in US Customary Units to SI units: {6}

(Rounded to the nearest millimeter. The input can actually be in any length unit, not just feet.)

Function:

$$\text{FT2M}(L_{ft}) := \text{Round}(L_{ft}, 1\ mm)$$

Examples:

$$\text{FT2M}(1234.56\ ft) = 376.294\ m$$

$$\text{FT2M}(6.5\ in) = 165\ mm$$

$$\text{FT2M}(1\ mi) = 1609.344\ m$$

$$\text{FT2M}(12.3456789\ m) = 12.346\ m$$

$$\text{FT2M}(33.8\ mm) = 0.034\ m$$

EXAMPLE CALCULATIONS

(a) Data Entered as Text Strings or Set of Three Scalar Values (FIF function) {7}

Mathcad's FIF function converts a string with FIF formatted text or a set of three scalar values into a dimensioned value, which can then be used in subsequent calculations. This is an easy way to convert raw FIF data into numbers. Unfortunately, with the text string, it's a bit confusing to have the inches symbol followed immediately by the double quote. As in the preceding section, Mathcad isn't fazed by oddball dimensions.

$L_{FIFstr\_1} := \text{“}10'7-9/25''\text{”}$

$L_{a\_1s} := FIF(10, 7, 0.36) = 127.360\ in$

$L_{a\_1} := FIF(L_{FIFstr\_1}) = 10.613333\ ft$

$L_{a\_1} = 127.360\ in$ FT2M $(L_{a\_1}) = 3.235\ m$

$$\text{FT2FIF}(L_{a\_1}) = \begin{bmatrix} 10 \\ 7 \\ 3 \\ 8 \end{bmatrix} \begin{bmatrix} \text{“ft”} \\ \text{“in”} \\ \text{“num”} \\ \text{“den”} \end{bmatrix}$$

$L_{FIFstr\_2} := \text{“}3'17-41/16''\text{”}$

$L_{a\_2s} := FIF(3, 17, 2.5625) = 55.5625\ in$

$L_{a\_2} := FIF(L_{FIFstr\_2}) = 4.630208\ ft$

$L_{a\_2} = 55.5625\ in$ FT2M $(L_{a\_2}) = 1.411\ m$

$$\text{FT2FIFA}(L_{a\_2}) = \begin{bmatrix} 4 & \text{“ft”} \\ 7 & \text{“in”} \\ 9 & \text{“num”} \\ 16 & \text{“den”} \end{bmatrix}$$

(b) Data Entered as a Simple Equation {7}

Another easy way to enter FIF data is to create a simple equation like the one below. Again, oddball dimensions are not a problem

$$L_{b\_1} := 2\ ft + \left(9 + \frac{5}{8}\right) in$$

$L_{b\_1} = 2.8021\ ft$

$L_{b\_1} = 33.6250\ in$ FT2M $(L_{b\_1}) = 854\ mm$

$$\text{FT2FIF}(L_{b\_1}) = \begin{bmatrix} 2 \\ 9 \\ 5 \\ 8 \end{bmatrix} \begin{bmatrix} \text{“ft”} \\ \text{“in”} \\ \text{“num”} \\ \text{“den”} \end{bmatrix}$$

$$L_{b\_2} := 3\ ft + \left(17 + \frac{41}{16}\right) in$$

$L_{b\_2} = 4.6302\ ft$

$L_{b\_2} = 55.5625\ in$ FT2M $(L_{b\_2}) = 1411\ mm$

$$\text{FT2FIFA}(L_{b\_2}) = \begin{bmatrix} 4 & \text{“ft”} \\ 7 & \text{“in”} \\ 9 & \text{“num”} \\ 16 & \text{“den”} \end{bmatrix}$$

(c) Data entered in FIF vectors and matrixes {8}

$$x1 := \begin{bmatrix} 0 \\ 0 \\ 6 \\ 16 \end{bmatrix} \quad x2 := \begin{bmatrix} 0 \\ 5 \\ 3 \\ 10 \end{bmatrix} \quad x3 := \begin{bmatrix} 1 \\ 0 \\ 7 \\ 4 \end{bmatrix} \quad x4 := \begin{bmatrix} 0 \\ 21 \\ 3 \\ 8 \end{bmatrix} \quad x5 := \begin{bmatrix} 0 \\ 15.25 \\ 0 \\ 1 \end{bmatrix} \quad x6 := \begin{bmatrix} 2 \\ 4 \\ 6 \\ 8 \end{bmatrix}$$

$$L_{c\_x1} := \mathrm{FIF2FT}(x1) = 0.031\ ft \quad L_{c\_x2} := \mathrm{FIF2FT}(x2) = 0.442\ ft \quad L_{c\_x3} := \mathrm{FIF2FT}(x3) = 1.146\ ft$$

$$\mathrm{FT2FIFA}(L_{c\_x1}) = \begin{bmatrix} 0 & \text{"ft"} \\ 0 & \text{"in"} \\ 3 & \text{"num"} \\ 8 & \text{"den"} \end{bmatrix} \quad \mathrm{FT2FIFA}(L_{c\_x2}) = \begin{bmatrix} 0 & \text{"ft"} \\ 5 & \text{"in"} \\ 5 & \text{"num"} \\ 16 & \text{"den"} \end{bmatrix} \quad \mathrm{FT2FIFA}(L_{c\_x3}) = \begin{bmatrix} 1 & \text{"ft"} \\ 1 & \text{"in"} \\ 3 & \text{"num"} \\ 4 & \text{"den"} \end{bmatrix}$$

$$L_{c\_x4} := \mathrm{FIF2FT}(x4) = 1.781\ ft \quad L_{c\_x5} := \mathrm{FIF2FT}(x5) = 1.271\ ft \quad L_{c\_x6} := \mathrm{FIF2FT}(x6) = 2.396\ ft$$

$$\mathrm{FT2FIFA}(L_{c\_x4}) = \begin{bmatrix} 1 & \text{"ft"} \\ 9 & \text{"in"} \\ 3 & \text{"num"} \\ 8 & \text{"den"} \end{bmatrix} \quad \mathrm{FT2FIFA}(L_{c\_x5}) = \begin{bmatrix} 1 & \text{"ft"} \\ 3 & \text{"in"} \\ 1 & \text{"num"} \\ 4 & \text{"den"} \end{bmatrix} \quad \mathrm{FT2FIFA}(L_{c\_x6}) = \begin{bmatrix} 2 & \text{"ft"} \\ 4 & \text{"in"} \\ 3 & \text{"num"} \\ 4 & \text{"den"} \end{bmatrix}$$

$$y1 := \begin{bmatrix} 0\ ft \\ 0\ in \\ 6\ num \\ 16\ den \end{bmatrix} \quad y2 := \begin{bmatrix} 0\ ft \\ 5\ in \\ 3\ num \\ 10\ den \end{bmatrix} \quad y3 := \begin{bmatrix} 1\ ft \\ 0\ in \\ 7\ num \\ 4\ den \end{bmatrix} \quad y4 := \begin{bmatrix} 0\ ft \\ 21\ in \\ 3\ num \\ 8\ den \end{bmatrix} \quad y5 := \begin{bmatrix} 0\ ft \\ 15.25\ in \\ 0\ num \\ 1\ den \end{bmatrix} \quad y6 := \begin{bmatrix} 2\ ft \\ 4\ in \\ 6\ num \\ 8\ den \end{bmatrix}$$

$$L_{c\_y1} := \mathrm{FIFU2FT}(y1) = 0.031\ ft \quad L_{c\_y2} := \mathrm{FIFU2FT}(y2) = 0.442\ ft \quad L_{c\_y3} := \mathrm{FIFU2FT}(y3) = 1.146\ ft$$

$$\mathrm{FT2FIFA}(L_{c\_y1}) = \begin{bmatrix} 0 & \text{"ft"} \\ 0 & \text{"in"} \\ 3 & \text{"num"} \\ 8 & \text{"den"} \end{bmatrix} \quad \mathrm{FT2FIFA}(L_{c\_y2}) = \begin{bmatrix} 0 & \text{"ft"} \\ 5 & \text{"in"} \\ 5 & \text{"num"} \\ 16 & \text{"den"} \end{bmatrix} \quad \mathrm{FT2FIFA}(L_{c\_y3}) = \begin{bmatrix} 1 & \text{"ft"} \\ 1 & \text{"in"} \\ 3 & \text{"num"} \\ 4 & \text{"den"} \end{bmatrix}$$

$$L_{c\_y4} := \mathrm{FIFU2FT}(y4) = 1.781\ ft \quad L_{c\_y5} := \mathrm{FIFU2FT}(y5) = 1.271\ ft \quad L_{c\_y6} := \mathrm{FIFU2FT}(y6) = 2.396\ ft$$

$$\mathrm{FT2FIFA}(L_{c\_y4}) = \begin{bmatrix} 1 & \text{"ft"} \\ 9 & \text{"in"} \\ 3 & \text{"num"} \\ 8 & \text{"den"} \end{bmatrix} \quad \mathrm{FT2FIFA}(L_{c\_y5}) = \begin{bmatrix} 1 & \text{"ft"} \\ 3 & \text{"in"} \\ 1 & \text{"num"} \\ 4 & \text{"den"} \end{bmatrix} \quad \mathrm{FT2FIFA}(L_{c\_y6}) = \begin{bmatrix} 2 & \text{"ft"} \\ 4 & \text{"in"} \\ 3 & \text{"num"} \\ 4 & \text{"den"} \end{bmatrix}$$

$$z1 := \begin{bmatrix} 0 & \text{"ft"} \\ 0 & \text{"in"} \\ 6 & \text{"num"} \\ 16 & \text{"den"} \end{bmatrix} \quad z2 := \begin{bmatrix} 0 & \text{"ft"} \\ 5 & \text{"in"} \\ 3 & \text{"num"} \\ 10 & \text{"den"} \end{bmatrix} \quad z3 := \begin{bmatrix} 1 & \text{"ft"} \\ 0 & \text{"in"} \\ 7 & \text{"num"} \\ 4 & \text{"den"} \end{bmatrix} \quad z4 := \begin{bmatrix} 0 & \text{"ft"} \\ 21 & \text{"in"} \\ 3 & \text{"num"} \\ 8 & \text{"den"} \end{bmatrix} \quad z5 := \begin{bmatrix} 0 & \text{"ft"} \\ 15.25 & \text{"in"} \\ 0 & \text{"num"} \\ 1 & \text{"den"} \end{bmatrix} \quad z6 := \begin{bmatrix} 2 & \text{"ft"} \\ 4 & \text{"in"} \\ 6 & \text{"num"} \\ 8 & \text{"den"} \end{bmatrix}$$

$$L_{c\_z1} := \mathrm{FIFA2FT}(z1) = 0.031\ ft \quad L_{c\_z2} := \mathrm{FIFA2FT}(z2) = 0.442\ ft \quad L_{c\_z3} := \mathrm{FIFA2FT}(z3) = 1.146\ ft$$

$$\mathrm{FT2FIFA}(L_{c\_z1}) = \begin{bmatrix} 0 & \text{"ft"} \\ 0 & \text{"in"} \\ 3 & \text{"num"} \\ 8 & \text{"den"} \end{bmatrix} \quad \mathrm{FT2FIFA}(L_{c\_z2}) = \begin{bmatrix} 0 & \text{"ft"} \\ 5 & \text{"in"} \\ 5 & \text{"num"} \\ 16 & \text{"den"} \end{bmatrix} \quad \mathrm{FT2FIFA}(L_{c\_z3}) = \begin{bmatrix} 1 & \text{"ft"} \\ 1 & \text{"in"} \\ 3 & \text{"num"} \\ 4 & \text{"den"} \end{bmatrix}$$

$$L_{c\_z4} := \mathrm{FIFA2FT}(z4) = 1.781\ ft \quad L_{c\_z5} := \mathrm{FIFA2FT}(z5) = 1.271\ ft \quad L_{c\_z6} := \mathrm{FIFA2FT}(z6) = 2.396\ ft$$

$$\mathrm{FT2FIFA}(L_{c\_z4}) = \begin{bmatrix} 1 & \text{"ft"} \\ 9 & \text{"in"} \\ 3 & \text{"num"} \\ 8 & \text{"den"} \end{bmatrix} \quad \mathrm{FT2FIFA}(L_{c\_z5}) = \begin{bmatrix} 1 & \text{"ft"} \\ 3 & \text{"in"} \\ 1 & \text{"num"} \\ 4 & \text{"den"} \end{bmatrix} \quad \mathrm{FT2FIFA}(L_{c\_z6}) = \begin{bmatrix} 2 & \text{"ft"} \\ 4 & \text{"in"} \\ 3 & \text{"num"} \\ 4 & \text{"den"} \end{bmatrix}$$

(d) Data Entered in a Table {9}

The following example mimics my Excel spreadsheet, the purpose of which is to add and subtract length dimensions and multiples of dimensions to produce a total distance. The data table can be lengthened or shortened as needed. The table includes the following columns:
COLUMN 1: Dim_Name provides a place to name each dimension for documentary purposes. This is optional and the cells can be left blank. In this example, the names represent the spacings between column grid lines on a building structural plan.
COLUMN 2: Qty is the number of times each dimension should be added (positive) or subtracted (negative) to produce the total length. In this example, certain spacings between column grid lines are repeated, which is common. The result of this calculation is the distance between Grid Line A and intermediate Grid Line J1, which we have to come back to from Grid Line K (Grid Lines J1 & J2 are between Grid Lines J & K). Qty need not be an integer. For example, the constant π can be used to determine the circumference of a circle from a dimension representing the diameter. Or, a trig function can be used (e.g. $sin(30°)$) for triangle solutions.
COLUMNS 3–6: FT, IN, NUM, and DEN, contains the feet, inch, numerator, and denominator values for each dimension. For dimensions that do not have a FT or an IN or a fraction (i.e. no NUM), enter zero in the appropriate column. For dimensions that do not have a fraction, DEN can be any value except 0, but I normally use 1. It not very common for grid line spacings to include fractions of an inch, and especially non-power-of-two fractions, but I included them here for demonstration purposes. Just like the preceding examples, the table method can deal with oddball dimensions just fine.

Dimensions to add/subtract to produce a total length:

Dim_Name	Qty	FT (ft)	IN (in)	NUM (in)	DEN
"A–B"	1	10	0	0	1
"B–C–D–E"	3	12	5	3	4
"E–F–G"	2	15	6	0	1
"G–H–I–J–K"	4	8	9	3	10
"H–J2–J1"	−2	2	6	5	32

Index values for data columns: $z := \text{length}(Qty) = 5$

$$i := a..(z+a-1)$$

Function to convert a row of table data to a dimension (multiply, then round):

$$L_{table}(Q, ft, in, num, den) := \text{Round}\left(Q \cdot \left(ft + in + \frac{num}{den}\right), \mathbf{Fraction}\right) \quad \{10\}$$

Function to convert a row of table data to a dimension (round, then multiply):

$$L_{table\_alt}(Q, ft, in, num, den) := Q \cdot \text{Round}\left(ft + in + \frac{num}{den}, \mathbf{Fraction}\right) \quad \{10\}$$

Convert data in Columns 3–6 to complete, factored length dimensions. {10}
(I prefer to round off AFTER multiplying by the quantity to preserve overall accuracy, so my follow-up calculations are based on the L_{table} function. If you prefer to round off before, use the L_{table_alt} function.)

$$L_{d_i} := L_{table}\left(Qty_i, FT_i, IN_i, NUM_i, DEN_i\right)$$

$$L_d = \begin{bmatrix} 120 \\ 449.25 \\ 372 \\ 421.1875 \\ -60.3125 \end{bmatrix} \, in$$

My upgraded HP-41CX program calculates using the "number of fractions" in each dimension, which is why I also show this result.

$$L_d = \begin{bmatrix} 1920 \\ 7188 \\ 5952 \\ 6739 \\ -965 \end{bmatrix} \, Fraction$$

$$L_{d\_alt_i} := L_{table\_alt}\left(Qty_i, FT_i, IN_i, NUM_i, DEN_i\right)$$

$$L_{d\_alt} = \begin{bmatrix} 120 \\ 449.25 \\ 372 \\ 421.25 \\ -60.375 \end{bmatrix} \, in$$

Note how the rounding order produces slight differences in dimensions that do not have fractions that are divisors of FR.

$$L_{d\_alt} = \begin{bmatrix} 1920 \\ 7188 \\ 5952 \\ 6740 \\ -966 \end{bmatrix} \, Fraction$$

Total length using the rounding after method: {11}

$$L_{d\_tot} := \sum_i L_{d_i} = 1302.1250 \, in$$

$$L_{d\_tot} = 108.51 \, ft \qquad \text{FT2M}\left(L_{d\_tot}\right) = 33.074 \, m$$

$$\text{FT2FIFA}\left(L_{d\_tot}\right) = \begin{bmatrix} 108 & \text{“ft”} \\ 6 & \text{“in”} \\ 1 & \text{“num”} \\ 8 & \text{“den”} \end{bmatrix}$$

GLOBAL DEFINITIONS

$\text{ORIGIN} \equiv 1$ $\qquad num \equiv 1$

$a \equiv \text{ORIGIN} = 1$ $\qquad den \equiv 1$

NOTES

{1a} I graduated from college in 1980, before SI became "really popular" :-) in the United States. Even so, I received sufficient grounding in the SI system in high school (Chemistry and Physics) and college (various engineering and science courses) to feel comfortable using it alongside the USCU system. In addition, I helped my dad design and build a house when I was 14 and got quite a workout with FIF dimensions and doing hand calcs with them. We had just moved in a few months before when my dad purchased the newly introduced HP-35 calculator. Even though the HP-35 (which I still have and which still works) isn't programmable (that would come later with the HP-65), it's too bad it wasn't available during design and construction.

{1b} I wrote my HP-41CX program so a drafter and I could work out various FIF calculations for about a dozen reinforced concrete structures I was designing and he was drafting (pre-Autocad) for a wastewater treatment plant in central California. In fact, he bought his own HP-41CX specifically to use my FIF program. Most of the calculations we had to do were simple additions and subtractions plus conversions back and forth with decimal feet. Our one triangle solution was for the Archimedes screw pumps that lifted the incoming sewage into the headworks. From the slope length between supports and the angle of inclination (provided by the manufacturer), we calculated the horizontal and vertical distances between supports.

{1c} I have been project manager and/or lead civil engineer for three federal prisons (and the QC reviewer for a major improvement project at a 4th), water projects at two military bases, and two buildings at another military base. Each project was fully SI* except for the building project. The building project was design-build and the Request for Proposals required the use of SI, as you would expect for a federal project. In addition, the supplied survey and geotechnical report used SI. Consequently, my project engineer and I developed our 30% designs using SI. Imagine our architect's surprise when we delivered our draft 30% designs to him. He and the contractor were unable to scale our drawings, so the architect called my project engineer and chewed him out. When my project engineer told him that our drawings used SI because that's what the RFP required, the architect had a fit about using SI, though he did apologize to my project engineer. Neither the contractor nor the architect had read the RFP in its entirety like we had and like they should have. The architect had never used SI, so he continued to work in USCU in spite of the RFP. This required my project engineer and me to make the conversions back to SI so we could fit his buildings onto the site. Somehow, with a proposal that used both SI (civil) and USCU (architectural), we were awarded the project. At the kickoff meeting, the architect asked the Navy's project manager if it would be OK to continue using USCU, rather than switch everything to SI. The Navy's project manager agreed to this request, remarking that neither he nor his staff liked SI anyway. At this point I raised my hand and requested that the Navy provide us with a survey in USCU, either a new survey or pay the surveyor-of-record to convert the original survey. He asked why I couldn't make the conversion and I told him that my boss would not accept the risk of converting someone else's survey. So, in the best tradition of Solomon, the Navy's project manager decreed that the site civil work would be done in SI and the building designs would be done in USCU. It was only because my project engineer and I knew what we were doing with both SI and USCU that these buildings even got built in the correct spots and to the correct overall dimensions.

*On all three prison projects, construction was a mish-mash of SI and back converted USCU, mainly due to individual construction and FBOP construction management personal being uncomfortable with SI. On one project, the underground subcontractor got on my case because he thought all of the larger storm drainage pipes we had designed would stick out of the ground. He had correctly figured out that a 1200-mm RCP soft converted back to a 48-inch RCP, but he hadn't figured out—in spite of numerous indications—that the survey and pipe inverts were in meters, not feet.)

{2} In addition to the common powers-of-two fractions ($FR =$ 2, 4, 8, 16,...), other useful numbers include 10 and 100. However, if you want to work in 25ths of an inch or 42nds of an inch or whatever, this worksheet will handle it.

{3} The GCD program code could just as easily have been included in the FT2FIF and FT2FIFA program codes that follows. However, I made it separate in case anyone wants an easy-to-grab program for GCD calculations.

{4} Two versions of this function program are provided. *FT2FIF* converts decimal feet into a 4-element unitless FIF vector, while *FT2FIFA* converts decimal feet into a 4x2 FIF matrix with units embedded as text in the second column. The inverse of *FT2FIF* is *FIF2FT* and the inverse of *FT2FIFA* is *FIFA2FT* {5}. Unfortunately, Mathcad does not have a way to use different actual units inside a results vector or matrix, so there is no *FT2FIFU* function program. However, the inverse of this non-existent function is provided (*FIFU2FT*) because it provides yet another style for entering data. In these two function programs, *Feet*, *Inches*, and *Numerator* calculate the three components of the FIF dimension, preserving the units for the subsequent steps. *FT*, *IN*, and *NUM*, strip off the units from these results for display purposes. *NUM* required an additional rounding because without it, some (but not all) GCD results were wonky. Change the rounding parameter to 15 and you will see what I mean. Dividing *NUM* and *FR* by the result of the GCD function produces the reduced fraction.

{5} These three functions convert FIF in vector/matric form into decimal feet. See {4} for additional details.

{6} Civil, structural, and architectural dimensions are typically specified to the nearest millimeter, although some structural elements my require greater precision.

{7} The simplest way to enter an FIF dimenion is with a properly formatted text string or a simple equation, both of which look a lot like the way we would write it. There is little to choose between the two and both methods seem to have their place. Although not shown, the equation method in (b) can also handle internal subtractions.

{8} These examples use the three functions described in {5} and one of the functions in {4}.

{9} For large calculations, I prefer the table approach to the three preceding methods. Even when I have the Autocad drawings, there are times when I want to permanently document an FIF calculation and this is how I would do it (or using the Excel spreadsheet).

{10} In these functions, $FIF\left(\frac{FT}{ft}, \frac{IN}{in}, \frac{NUM}{in \cdot DEN}\right)$ could be used instead of $\left(ft + in + \frac{num}{den}\right)$. Sometimes (though rarely for me) structural elements are dimensioned in 10ths or 100ths of an inch. When such data is provided, but final results are required in, say, sixteenths of an inch, it is best to multiply first, then round to preserve overall accuracy. In reality, most dimensions are provided using standard fractions that are no smaller than the chosen fraction, which makes all this a moot point for just about every situation.

{11} This step calculates subtotal dimensions by table row using the values in the five numerical data columns in the table. For Mathcad purposes, I prefer to display these intermediate results in inches. The second version of these intermediate results shows how many "fractions" each row amounts to, which is how my upgraded HP-41CX program handles each dimension internally. For example, ROW1 totals 10 ft or 120 inches or 1920 fractions (i.e. sixteenths of an inch). If the *FR* is changed to 4, then ROW1 becomes 480 fractions.

{12} The final step sums the subtotals and displays the result in decimal feet, in meters (rounded to the nearest millimeter by the FT2M function), and in FIF vector format (including a rounded and reduced fraction).

REFERENCES

[1] Euclidean algorithm for the Greatest Common Divisor: http://en.wikipedia.org/wiki/Euclidean_algorithm.

[2] Nelson, Richard J., et al, *PPC ROM User's Manual*, pp. 170–173, PPC Inc., Santa Ana, CA (1981).