PROGRAMMING

quicksheets
Looping and Control

This QuickSheet provides some examples of using **For** and **While** loops to iterate calculations, and demonstrates conditional program control using the **if** and **continue** statements. While looping and iteration is implied using range variables and indexes for vectors, it may be more efficient, or clearer to write things as a Mathcad program.

Copy each program into a blank worksheet, turn on Debug mode, and add **trace** or **pause** functions so that you can watch the values of the local variables update as the loops progress. See **Program Debugging** in the Tutorials.

For loops range over values specified explicitly in the loop instruction. For example, this program flattens a matrix into a column vector, starting with the last column.

$$M := \mathrm{identity}(3)$$

```
flatten(M) := | $v \leftarrow M^{\langle \mathrm{cols}(M)-1 \rangle}$
              | for $j \in \mathrm{cols}(M) - 2 .. 0$  if $\mathrm{cols}(M) > 1$
              |     $v \leftarrow \mathrm{stack}\left(v, M^{\langle j \rangle}\right)$
              | $v$
```

$$\mathrm{flatten}(M)^{T} = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

The value of M is passed as an argument to a function, which is the most reusable way to write programs in Mathcad. You can refer to worksheet variable names directly as well. If you name a local variable by the same name, it refers to the worksheet variable only until it is initialized.

$a := 5$

```
reuse :=  | for  $i \in 0..3$
          |     $a \leftarrow a + i$
          | $a$
```

$$\sum_{i=0}^{3} i = 6$$

$reuse = 11$

After the first iteration, *a* is the local variable, not the worksheet value. Be aware of this if you are renaming variables or functions within loops.

Continue Operator

The continue operator stops execution of the current iteration, and restarts it at the top of the nearest enclosing loop for the next iteration. For example, this program adds only odd integers between 0 and *n* by skipping the even ones using **continue**.

```
addOdd(n) :=  | $s \leftarrow 0$
              | for  $k \in 0..n$
              |     | continue   if  $\text{mod}(k,2) = 0$
              |     | $s \leftarrow s + k$
              | $s$
```

$addOdd(10) = 25$ $\qquad$ $1 + 3 + 5 + 7 + 9 = 25$

By comparison, the summation operator includes all nonnegative integers.

$$\sum_{j=0}^{10} j = 55$$

While Loops

While loops terminate when a condition is first violated. This program finds the square root of a positive real number, terminating when the estimate of the root squared is less than a threshold amount different from target value.

```
sqrt(a, ε) :=  | estsqrt ← 1
               | while $\left|estsqrt^2 - a\right| \ge \varepsilon$
               |   estsqrt ← $\frac{1}{2}\cdot\left(estsqrt + \frac{a}{estsqrt}\right)$
```

$$\mathrm{sqrt}\left(37, 1\times 10^{-5}\right) = 6.082762537585$$

Find the first element in a vector that exceeds a certain threshold.

$$j := 0..2.5\times 10^{3} \quad v_j := \sin(j)$$

```
t(v, thres) :=  | j ← 0
                | while $\left|v_j\right| \le thres$
                |   j ← j + 1
                | $\begin{pmatrix} j \\ v_j \end{pmatrix}$
```

$$t(v, 0.99999) = \begin{pmatrix} 11 \\ -0.99999021 \end{pmatrix}$$

Break Operator

You can break out of a **while** or **for** loop prematurely using the **break** operator. This program refines the estimate of the square root until it is better than e, or until it has reached its maximum number of iterations.

```
$\mathrm{sqroot}(a, \varepsilon, \mathrm{maxiter}) :=$
    $\mathrm{estsqrt} \leftarrow 1$
    for $i \in 0 .. \mathrm{maxiter}$
        $\mathrm{estsqrt} \leftarrow \frac{1}{2} \cdot \left(\mathrm{estsqrt} + \frac{a}{\mathrm{estsqrt}}\right)$
        if $\left|\mathrm{estsqrt}^2 - a\right| < \varepsilon$
            $\mathrm{maxiter} \leftarrow i$
            break
    $\begin{pmatrix} \mathrm{estsqrt} \\ \mathrm{maxiter} \end{pmatrix}$
```

$$\mathrm{sqroot}\left(3.057 \times 10^{3}, 10^{-6}, 100\right) = \begin{pmatrix} 55.290143787116 \\ 9 \end{pmatrix}$$

Defining an Iterative Function

Below is a factorial function, equivalent to Mathcad's built-in factorial operator.

```
$\mathrm{Fac}(n) :=$
    $f \leftarrow 1$
    while $n \leftarrow n - 1$
        $f \leftarrow f \cdot (n + 1)$
    $f$
```

$$\mathrm{Fac}(6) = 720$$

The **while** loop checks if a condition is true or false, with false equal to 0. When the value of *n* is assigned to 0, the loop terminates.