```
path := CWD                    file := "Parameter_Sheet.xlsx"
```

Use the standard definition assignment

```
Mat := READEXCEL(concat(path, "\" , file) , "Material!" )

Coeff := READEXCEL(concat(path, "\" , file) , "Coefficients!" )
```

Reading parameters from Excel file

The symbolic processor sees the functions READEXCEL and concat, but doesn't know how to evaluate them. Evaluating Mat and Coeff results in the simple substitutions.

```
Mat → READEXCEL(as_string("\\USER-PC\Mathcad\Collaboratory\") , concat("\\USER-PC\Mathcad\Co
Coeff → READEXCEL(as_string("\\USER-PC\Mathcad\Collaboratory\") , concat("\\USER-PC\Mathcad\(
```

The symbolic processor doens't handle string functions, so that's one are we can try to get round by removing the concat ....

evaluate the complete file name, copy it and paste directly into READEXCEL

```
concat(path, "\" , file) = "\\USER-PC\Mathcad\Collaboratory\\Parameter_Sheet.xlsx"

Mat := READEXCEL("\\USER-PC\Mathcad\Collaboratory\\Parameter_Sheet.xlsx" , "Material!" )

Coeff := READEXCEL("\\USER-PC\Mathcad\Collaboratory\\Parameter_Sheet.xlsx" , "Coefficients!" )
```

This gets rid of the concat in the symbolic evaluation, but demonstrates that the symbolic processor can't handle READEXCEL.

```
Mat → READEXCEL(as_string("\\USER-PC\Mathcad\Collaboratory\") , "\\USER-PC\Mathcad\Collaborat
Coeff → READEXCEL(as_string("\\USER-PC\Mathcad\Collaboratory\") , "\\USER-PC\Mathcad\Collabor
```

One way round this is to in-line evaluate Mat and Coeff. From a symbolic processor perspective, this assigns the values of the evaluation to Mat and Coeff rather than assigning READEXCEL.

```
                                                  ⎛          "What"                 "Value"    
                                                  ⎜        "Material"           "AW-5083 H112"
Mat := READEXCEL(concat(path, "\" , file) , "Material!" ) =  ⎜ "Tensile strength RmM"       290       
                                                  ⎝  "Yield Strength Rp02"         150       

                                                          ⎛                "What"
                                                          ⎜ "Safety factor tensile strength S1"
Coeff := READEXCEL(concat(path, "\" , file) , "Coefficients!" ) = ⎜  "Safety factor yield strength S2"
                                                          ⎝"Safety factor during pressure test S3"
```

The symbolic processor now evaluates to the data rather than the functional expression to read the data.

$$\text{Mat} \rightarrow \begin{pmatrix} \text{"What"} & \text{"Value"} & \text{"Unit"} \\ \text{"Material"} & \text{"AW-5083 H112"} & \text{"-"} \\ \text{"Tensile strength RmM"} & 290 & \text{"N/mm2"} \\ \text{"Yield Strength Rp02"} & 150 & \text{"N/mm2"} \end{pmatrix}$$

$$\text{Coeff} \rightarrow \begin{pmatrix} \text{"What"} & \text{"Value"} & \text{"Unit"} \\ \text{"Safety factor tensile strength S1"} & 3.5 & \text{"-"} \\ \text{"Safety factor yield strength S2"} & 1.8 & \text{"-"} \\ \text{"Safety factor during pressure test S3"} & 1.1 & \text{"-"} \end{pmatrix}$$

$$\text{Mat} := \text{submatrix}(\text{Mat}, 1, \text{rows}(\text{Mat}) - 1, 1, 1)$$

$$\text{Coeff} := \text{submatrix}(\text{Coeff}, 1, \text{rows}(\text{Coeff}) - 1, 1, 1)$$

Truncating array

$$\text{Mat} \rightarrow \begin{pmatrix} \text{"AW-5083 H112"} \\ 290 \\ 150 \end{pmatrix} \qquad \text{Coeff} \rightarrow \begin{pmatrix} 3.5 \\ 1.8 \\ 1.1 \end{pmatrix}$$

Unfortunately, this still doesn't resolve the explicit problem, as the symbolic processor doesn't appear to like having directly indexed matrix elements in the evaluation.

Some calculation with display of used values (for better oversight)

Obviously the explicit command reaches back the entire path even when not invoking the keyword ALL.
Is there a way ti limit the number of steps back for the explicit function?

$$R_m := \frac{\text{Mat}_1}{\text{Coeff}_0} \rightarrow \; =$$

$$R_m := \frac{m1}{c0} \rightarrow \frac{\text{Mat}_1}{\text{Coeff}_0} = 82.857$$

assigning the values to variables pulls up the source matrix elements when using explicit

$$m1 := \text{Mat}_1$$

$$c0 := \text{Coeff}_0$$

$$R_m := \frac{m1}{c0} \rightarrow \frac{\text{Mat}_1}{\text{Coeff}_0} = 82.857$$

However, using in-line evaluation does work ... bit of a long-winded method though!

$$m1 := \text{Mat}_1 = 290$$

$$c0 := \text{Coeff}_0 = 3.5$$

$$R_{m} := \frac{m1}{c0} \rightarrow \frac{290}{3.5} = 82.857$$

$\text{mat} := \text{stack}(1, 3, 5)$ $\text{coeff} := \text{stack}(5, 3, 1)$

$$\text{rm} := \frac{\text{mat}_{1}}{\text{coeff}_{1}} \ \text{explicit}, \text{mat}_{1}, \text{coeff}_{1} \rightarrow \quad \frac{\text{mat}_{1}}{\text{coeff}_{1}} \ \text{explicit}, \text{all} \rightarrow \frac{\text{mat}_{1}}{\text{coeff}_{1}}$$

$Mat := Mat \qquad Coeff := Coeff$

llaboratory\" , "\" , "Parameter_Sheet.xlsx") , "Material!")

:ollaboratory\" , "\" , "Parameter_Sheet.xlsx") , "Coefficients!")

:ory\\Parameter_Sheet.xlsx" , "Material!")

:atory\\Parameter_Sheet.xlsx" , "Coefficients!")

$$\begin{matrix} \text{"Unit"} \\ \text{"-"} \\ \text{"N/mm2"} \\ \text{"N/mm2"} \end{pmatrix}$$

$$Mat = \begin{pmatrix} \text{"What"} & \text{"Value"} \\ \text{"Material"} & \text{"AW-5083 H112"} \\ \text{"Tensile strength RmM"} & 290 \\ \text{"Yield Strength Rp02"} & 150 \end{matrix}$$

$$\begin{matrix} \text{"Value"} & \text{"Unit"} \\ 3.5 & \text{"-"} \\ 1.8 & \text{"-"} \\ 1.1 & \text{"-"} \end{pmatrix}$$

$$Coeff = \begin{pmatrix} \text{"What"} & \text{"V} \\ \text{"Safety factor tensile strength S1"} & \\ \text{"Safety factor yield strength S2"} & \\ \text{"Safety factor during pressure test S3"} & \end{matrix}$$

$$\text{Mat} = \begin{pmatrix} \text{"AW-5083 H112"} \\ 290 \\ 150 \end{pmatrix} \qquad \text{Coeff} = \begin{pmatrix} 3.5 \\ 1.8 \\ 1.1 \end{pmatrix}$$

$\text{Mat}_1 = 290$ $\qquad$ $\text{Coeff}_0 = 3.5$

$$\left.\begin{array}{c}\text{"Unit"}\\ \text{"-"}\\ \text{"N/mm2"}\\ \text{"N/mm2"}\end{array}\right)$$

$$\left.\begin{array}{ll}\text{alue"} & \text{"Unit"}\\ 3.5 & \text{"-"}\\ 1.8 & \text{"-"}\\ 1.1 & \text{"-"}\end{array}\right)$$