
1. PROBLEM DESCRIPTION

Currently, LibreOffice Math equations will not render incomplete parts. This is most painful when writing equations with large matrices, since the live-preview is effectively disabled until writing the matrix is finished.

For other structures, writing some scaffolding before filling in the contents can help, for instance writing

```
left( right)
```

before filling in the contents between the parentheses. But for a matrix, writing

```
left(matrix{
  m_1' E # 0 # 0 # 0 # dotsaxis ##
  0 # I_1' # 0 # 0 # dotsaxis ##
  0 # 0 # m_2 E
}right)
ℒ
```

still won't render, as matrices are only rendered when all rows have the same length.

2. SUGGESTED ENHANCEMENT

Instead it could be rendered as

```
left(matrix{
  m_1' E # 0      # 0      # 0      # dotsaxis ##
  0      # I_1' # 0      # 0      # dotsaxis ##
  0      # 0      # m_2 E # <?> # <?>
}right)
⎛ m1'E   0    0    0    ... ⎞
⎜  0    I1'   0    0    ... ⎟
⎝  0    0   m2E  □    □ ⎠
```

and something like

```
left(matrix{
  m_1' E # 0      # 0      # 0      # dotsaxis ##
  0      # I_1' # 0      # 0      # dotsaxis ##
}right)
ℒ
```

could be rendered as either

```
left(matrix{
  m_1' E # 0      # 0 # 0      # dotsaxis ##
  0      # I_1' # 0 # 0      # dotsaxis
}right)
⎛ m1'E   0    0    0    ... ⎞
⎜  0    I1'   0    0    ... ⎟
```

or

```

left(matrix{
  m_1' E # 0      # 0      # 0      # dotsaxis ##
  0      # I_1' # 0      # 0      # dotsaxis ##
  <?>    # <?>    # <?>    # <?>    # <?>
}right)

```

$$\begin{pmatrix} m_1' E & 0 & 0 & 0 & \dots \\ 0 & I_1' & 0 & 0 & \dots \\ \square & \square & \square & \square & \square \end{pmatrix}$$

3. WORKAROUND AND WHY IT IS INSUFFICIENT

This *can* be solved by preparing a scaffolding

```

left(matrix{
  . # . # . # . # . ##
  . # . # . # . # . ##
  . # . # . # . # . ##
  . # . # . # . # . ##
  . # . # . # . # .
}right)

```

$$\begin{pmatrix} . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \end{pmatrix}$$

but is very inconvenient, and doesn't work if during writing it turns out that an additional column has to be added, especially in the middle. Additionally, creating such a large scaffolding structure breaks the flow of writing a lot more than a simpler construct like `left(right)`.

The latter scaffoldings in many cases are even already supported as keyboard shortcuts, such as

Shift+F10, B, H

to insert

```

left( <?> right)

```

$$\square$$

and select the `<?>` placeholder. No such thing exists for matrices, for which a whole wizard would be needed.

4. ADVANTAGE OF HANDLING SPECIAL CASE OF MATRIX

Similar issues arise for many other constructs. While eliminating the need for a "scaffolding first" style of writing equations would be desirable, it would result in unclear situations, especially when editing in the middle of an equation.

For instance,

$$f(t) = \sin(a +$$

$$f(t) = \sin(\textcolor{red}{t})$$

could reasonably be rendered as

$$f(t) = \sin(a + \text{<?>})$$

$$f(t) = \sin(a + \square)$$

but when encountering

$$f(t) = \sin(a + \cos(c + d)$$

$$f(t) = \sin(\textcolor{red}{t})$$

its not clear if rendering this as

$$f(t) = \sin(a + \cos(c + d))$$

$$f(t) = \sin(a + \cos(c + d))$$

would be a desirable change.

By contrast, filling up a matrix with <?> until it is a complete rectangular matrix affects only the rendering of one component of the equation, and will at worst result in shifting expression from one column to the next, when another column is added in the middle.