



# TRAMETOOLS



## TRAMECALC Metalanguage — Specification

The TRAMECALC Metalanguage is a lightweight, text-based language designed for structured engineering calculations. It integrates user input, symbolic/numeric computation (via Octave), and high-quality mathematical rendering using Maxima and MathJax.



### General Concept

A TRAMECALC file defines:

Input parameters

Mathematical expressions

Execution logic (via Octave)

Structured report output (HTML)

The language is interpreted within an Octave environment, where all variables become globally accessible.



# TRAMETOOLS

## Syntax Overview

### Comments (Hashbang style)

# **This is a comment**

Used for documentation and ignored during execution.

### Image Inclusion

@ **logo.png**

Displays an image located in the calculation folder within the HTML report.

### Title

% **Title of the calculation**

Defines the main title of the report.

### Horizontal Line

---

Inserts a horizontal separator in the HTML report.

◆ **Subtitle**

' **Subtitle**

Adds a subsection title, following the logical sequence of the calculation.

## TRAMECALC Metalanguage

### # Text-based blocks

```
@ logo.png
% Calculation Title
---
' Geometry
L : 10 m Length
W : 2.5 m Width
F = 4*L kgf Load
```



### @ Include image

```
% Set title
' Add subtitle
var : value Input variables
var = expr Calculate results
>> octave Run Octave code
```

Octave

GNU PLOT

MathJax

Maxima

Structured calculation reports

GNU PLOT

Structured calculation reports

```
@ Include image
% Set title
' Add subtitle
var = expr Calculate results
var : value Input variables
>> octave Run Octave code
* string Print string
```

GNU PLOT

GNU PLOT

GNU PLOT

GNU PLOT

GNU PLOT

GNU PLOT

GNU PLOT

GNU PLOT

GNU PLOT

GNU PLOT

GNU PLOT

GNU PLOT

GNU PLOT

GNU PLOT

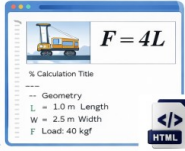
GNU PLOT

GNU PLOT

GNU PLOT

GNU PLOT

GNU PLOT



Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports

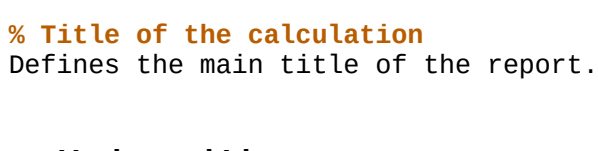
Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports



Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports

Structured calculation reports



# TRAMETOOLS

## Variable Definitions

### ✓ Input Variable

**var** : value unit description

Defines an initial value for a variable  
The user will be prompted (via TUI) to confirm or modify this value at runtime

Example:

**L** : 5 m Beam length

### ✓ Calculated Variable

**var** : expression unit descriptio

Evaluates a mathematical expression  
Stores the result in var  
The expression is rendered in the HTML report using Maxima + MathJax

Example:

**A** =  $b \cdot h$  cm<sup>2</sup> Cross-sectional area



# TRAMETOOLS

## HTML Integration ✓ Inline Execution

`<>` `< HTML code >`

Executes inline HTML code.

Your code is written in the HTML report.

Example:

```
<> 
```



## Octave Integration

### ✓ Inline Execution

`>>` `(octave_command)`

Executes inline Octave code.

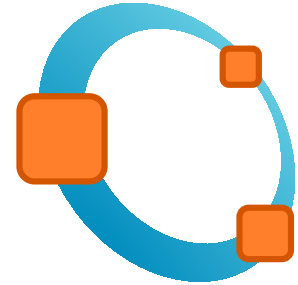
Runs inside the same session.

All variables are global.

Can define functions, intermediate values, etc.

Example:

```
>> I = b*h^3/12;
```



## String Output

### ✓ Print String Variable

\* `string_variable`

Outputs the content of a string variable into the HTML report.



# TRAMETOOLS

Editing the calc file on the Nano editor.

```
GNU nano 7.2 metacalc
# Input an image
@ Logo.png

# Title
% Metalanguage explained

# Horizontal line
---

#Subtitle
' INPUT

#Define variables - automatic TUI input
sigma : 25 MPa Bendind stress
tau : 30 MPa Shear stress
Fr : 50 MPa Stress Strength

' OUTPUT
Fs = sqrt(sigma^2+tau^2) MPa Maximum stress
ratio = Fs/Fr . Ratio

# Octave commands
>> if ratio <= 1; check='Ok!'; else check = 'Not Ok!'; end;

* check

^G Help      ^O Write Out  ^W Where Is   ^K Cut
^X Exit      ^R Read File  ^\ Replace    ^U Paste
```

Mozilla Firefox  
/home/paulo/trametools/calc/E x +  
file:///home/paulo/trametools/calc/Examples 120% ☆

## TRAMETOOLS

### Metalanguage explained

---

#### INPUT

$\sigma = 25$  MPa - Bendind stress  
 $\tau = 30$  MPa - Shear stress  
 $Fr = 50$  MPa - Stress Strength

#### OUTPUT

Maximum stress

$$Fs = \sqrt{\tau^2 + \sigma^2} = 39.0512 \text{ MPa}$$

Ratio

$$ratio = \frac{Fs}{Fr} = 0.781025 .$$

Ok!



# TRAMETOOLS

## Execution Model

The file is parsed sequentially

Variables are defined or computed

Octave executes embedded commands

Expressions are processed via Maxima

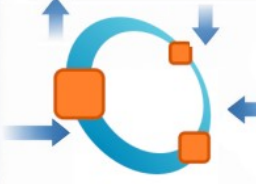
Final output is rendered in HTML using MathJax for equations and GNUPlot for graphics.

## TRAMECALC Metalanguage

# Text-based blocks

```
@ logo.png
% Calculation Title
---
' Geometry
L : 10 m Length
W : 2.5 m Width
F = 4*L kgf Load
```

```
@ Include image
% Set title
' Add subtitle
var = expr Calculate results
var : value Input variables
>> octave Run Octave code
* string Print string
```



@ Include image

% Set title

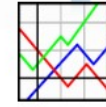
' Add subtitle

var : value Input variables

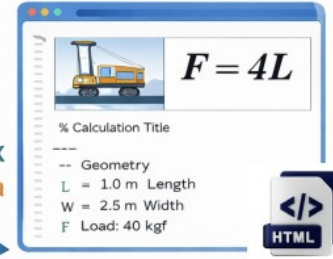
var = expr Calculate results

>> octave Run Octave code

Octave



GNU PLOT



Structured calculation reports



**TRAMETOOLS**

**Thank You !**