



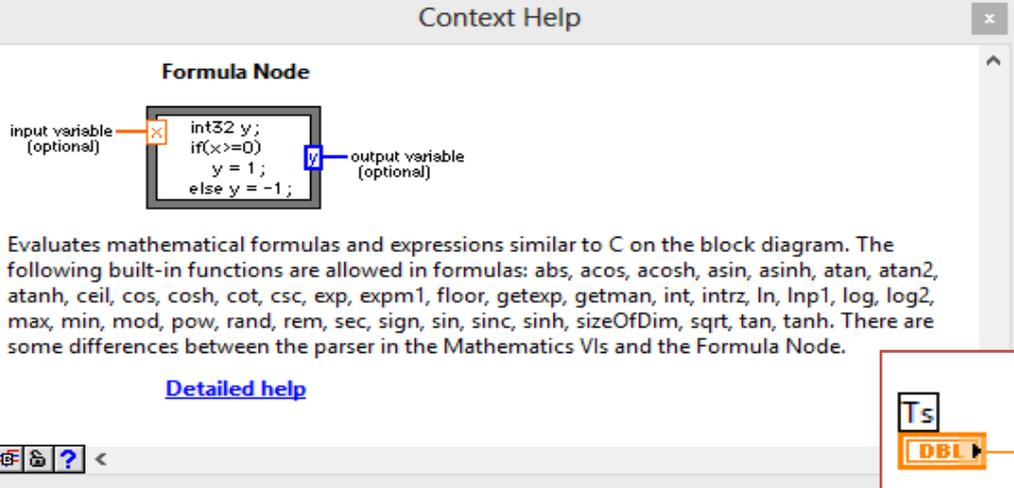
LabVIEW Formula Node

Hans-Petter Halvorsen

LabVIEW Formula Node

Formula Node: Create and use C code within LabVIEW

Example:

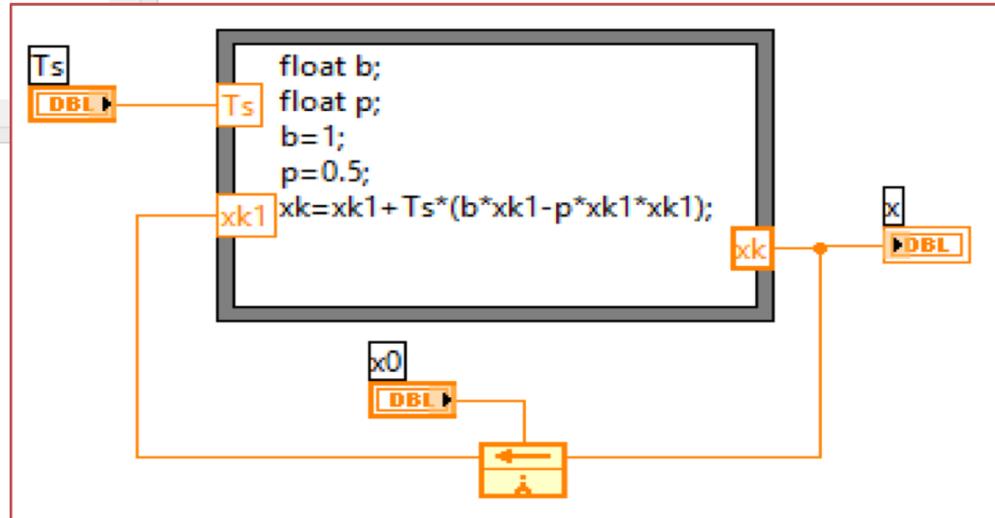


The screenshot shows the 'Context Help' window for the 'Formula Node'. The title bar reads 'Context Help'. The main content area is titled 'Formula Node' and contains a diagram of the node. On the left, an orange arrow labeled 'input variable (optional)' points to a small square icon with an 'x'. On the right, a blue arrow labeled 'output variable (optional)' points to a small square icon with a 'y'. The central box contains the following C code:

```
int32 y;  
if(x>=0)  
    y = 1;  
else y = -1;
```

Below the diagram, a paragraph explains that the node evaluates mathematical formulas and expressions similar to C. It lists built-in functions: `abs`, `acos`, `acosh`, `asin`, `asinh`, `atan`, `atan2`, `atanh`, `ceil`, `cos`, `cosh`, `cot`, `csc`, `exp`, `expm1`, `floor`, `getexp`, `getman`, `int`, `intrz`, `ln`, `lnp1`, `log`, `log2`, `max`, `min`, `mod`, `pow`, `rand`, `rem`, `sec`, `sign`, `sin`, `sinc`, `sinh`, `sizeofDim`, `sqrt`, `tan`, `tanh`. It also notes that there are differences between the parser in the Mathematics VIs and the Formula Node. A blue link labeled 'Detailed help' is provided. At the bottom left, there are icons for home, back, and help.

Very useful for mathematical expressions and simulations (implementing discrete equations)!



LabVIEW Formula Node Example

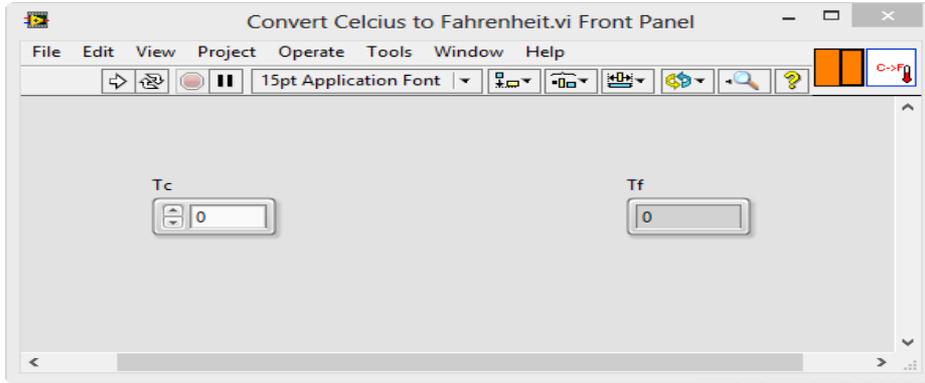
Celsius to Fahrenheit:

$$T_F = \frac{9}{5} T_C + 32$$

We will use the LabVIEW Formula Node in order to implement this formula

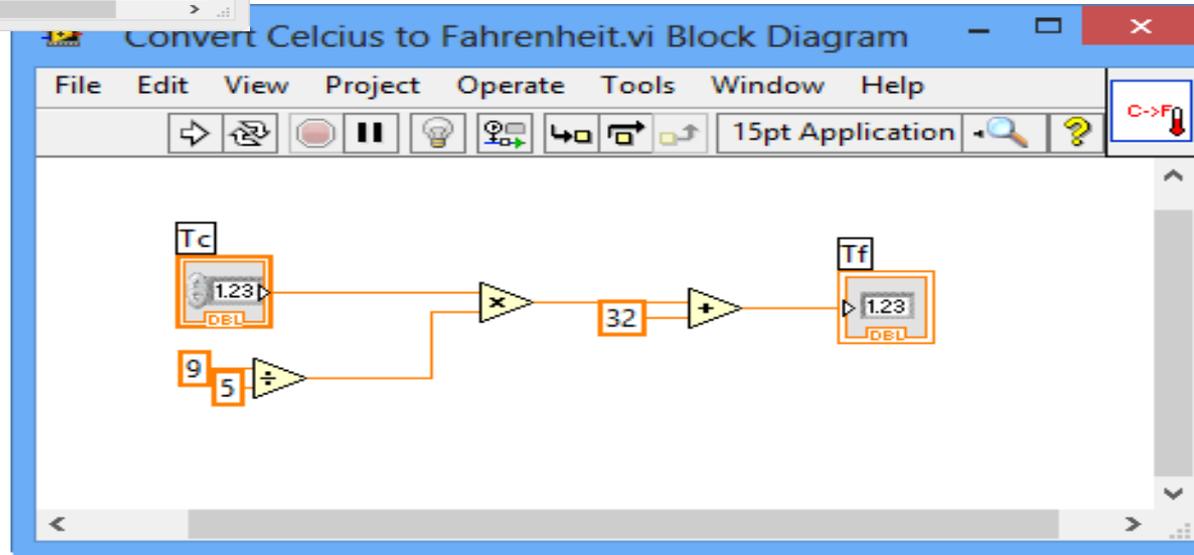
Celsius to Fahrenheit - Example

Front Panel



$$T_F = \frac{9}{5} T_C + 32$$

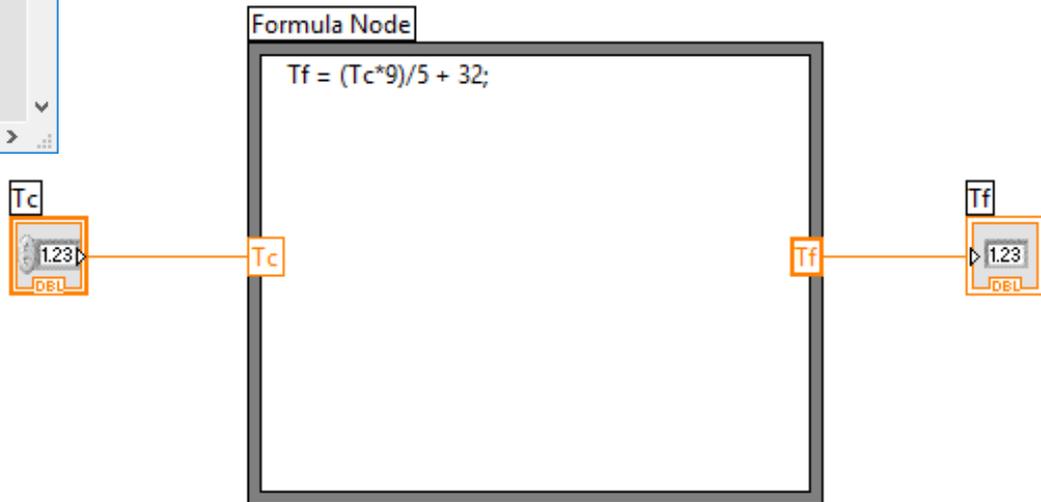
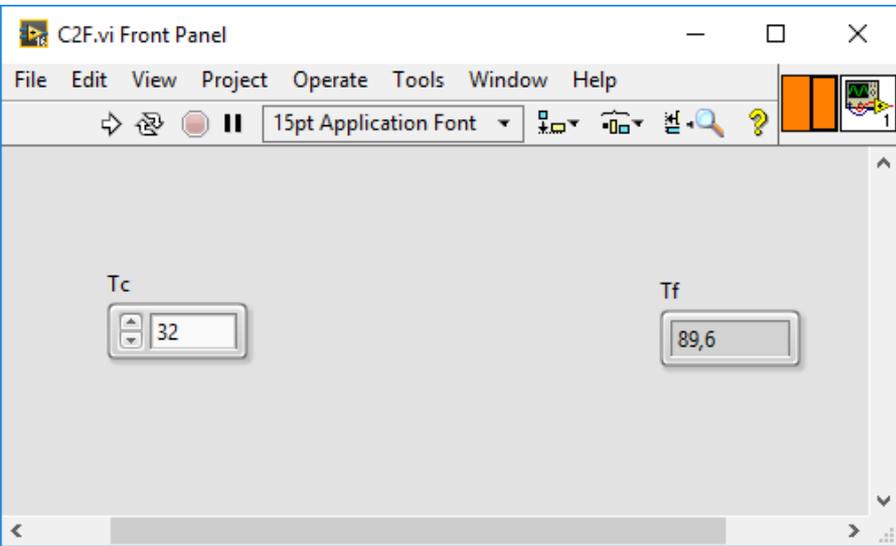
Block Diagram



Here we have used ordinary LabVIEW functionality.

Lets try to use the Formula Node instead.

Celsius to Fahrenheit - Example



DEMO

LabVIEW Formula Node

Advanced Mathematical Formula:

$$f(x) = \frac{\ln(ax^2 + bx + c) - \sin(ax^2 + bx + c)}{4\pi x^2 + \cos(x - 2)(ax^2 + bx + c)}$$

Given $a = 1, b = 3, c = 5$

Find $f(9)$

We will use the LabVIEW Formula Node in order to implement this formula

(The answer should be $f(9) = 0.0044$)

TRY IT OUT!

Solutions

$$f(x) = \frac{\ln(ax^2 + bx + c) - \sin(ax^2 + bx + c)}{4\pi x^2 + \cos(x - 2)(ax^2 + bx + c)}$$

Formula Node Example.vi Front Panel

`f = (ln(a*x*x + b*x + c) - sin(a*x*x + b*x + c)) / (4*pi*x*x + cos(x-2)*(a*x*x + b*x + c));`

Note!

Inputs: a, b, c, x

Output: f

Formula Node Example.vi Front Panel

Inputs: a=1, b=3, c=5, x=9

Output: f=0,00437378

Formula Node Example2.vi Block Diagram

Alternative Solution:

```
float g;  
g = a*x*x + b*x + c;  
f = (ln(g) - sin(g)) / (4*pi*x*x + cos(x-2)*(g));
```

Which Solution do you think is best?

Inputs: a, b, c, x

Output: f

LabVIEW MathScript Node and LabVIEW MATLAB Node

- They use MATLAB Syntax
- More powerful features regarding Arrays and Matrices
- You can use hundreds/thousands of built-in functions
- MathScript Node:
 - You need to have LabVIEW MathScript RT Module installed
 - MathScript is an add-on to LabVIEW, and it includes a “miniature” version of MATLAB
- MATLAB Node:
 - You need to have MATLAB installed on your computer
 - Works only for Windows (it uses ActiveX)

LabVIEW MathScript Window

The screenshot displays the LabVIEW MathScript environment. The main window is titled "LabVIEW MathScript" and contains an "Output Window" on the left and a "Variables" panel on the right. The "Output Window" shows the following commands and their results:

```
>>clear
>>x=0:10
x =
    0     1     2     3     4     5     6
    7     8     9    10    11    12    13
>>y=2*x+1
y =
    1     3     5     7     9    11    13
    15    17    19    21    23    25    27
>>plot(x,y)
```

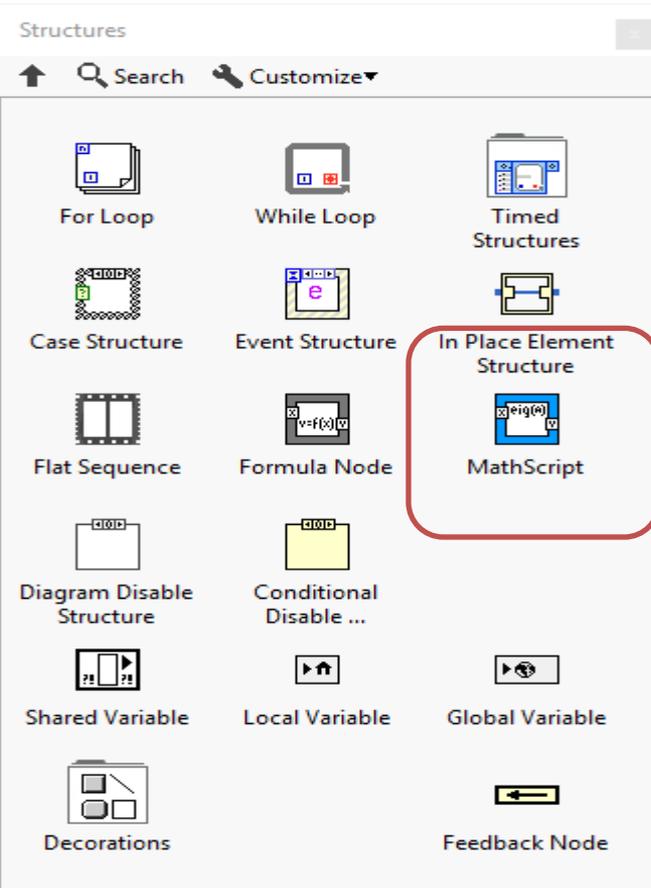
The "Variables" panel shows the following table:

Partition/Variable	Data	Type
Global		
Local		
x	<1x11>	int32 array
y	<1x11>	int32 array

A "Plot 1" window is overlaid on the main window, showing a graph of the function $y = 2x + 1$. The x-axis ranges from 0 to 10, and the y-axis ranges from 0 to 22. The plot shows a blue line starting at (0, 1) and ending at (10, 21).

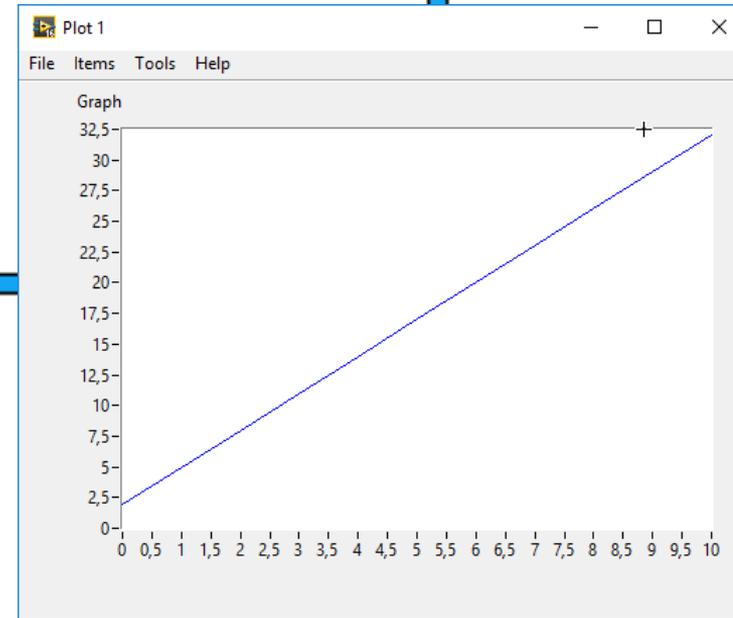
At the bottom of the LabVIEW window, the version number "16.0" and the status "Idle" are visible.

LabVIEW MathScript Node

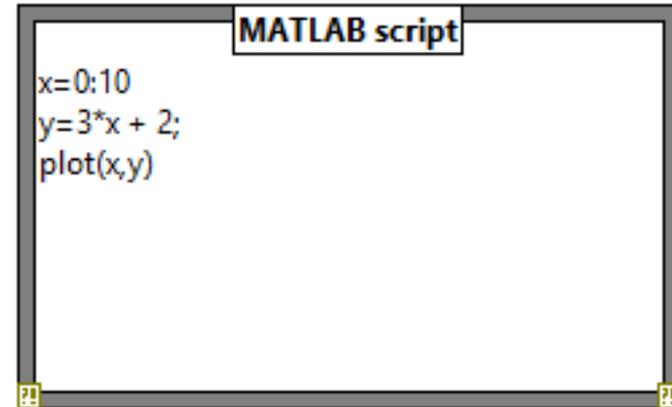
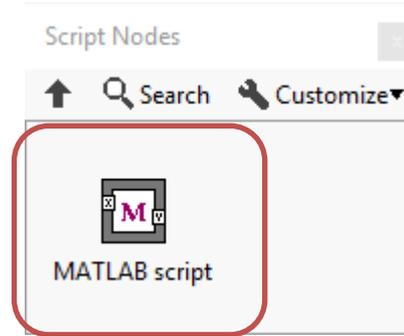
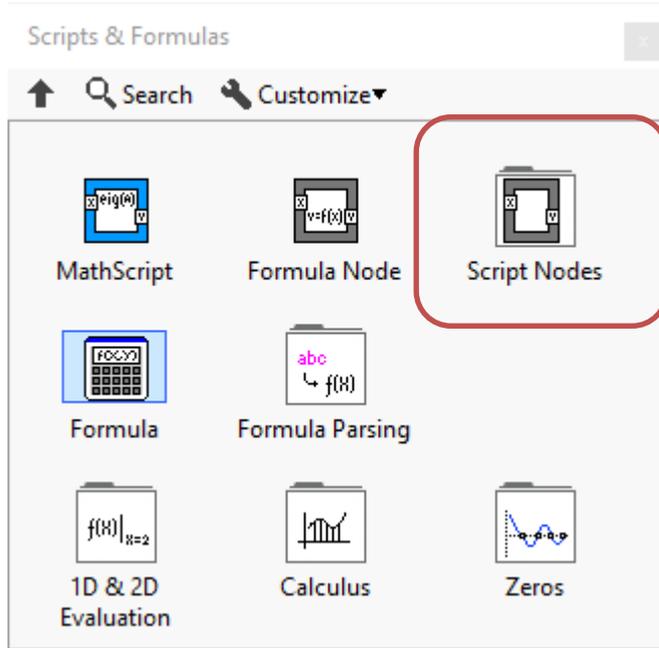


MathScript Node

```
1 x = 0 : 10;  
2  
3 y = 3*x + 2;  
4  
5 plot(x,y)  
6
```



LabVIEW MATLAB Node



DEMO

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: <https://www.halvorsen.blog>

