SLang - the Next Generation

Tutorial

Christian Bucher, Sebastian Wolff
Center of Mechanics and Structural Dynamics
Vienna University of Technology

November 8, 2010

0.1 Simple mathematical functions

Let us assume that we would like to compute the functions $f_k(x)$ in the interval $x \in [0, 40]$. The functions are $f_1(x) = \sin(0.2 \cdot x)$, $f_2(x) = \cos(0.12 \cdot x)$ and $f_3(x) = \exp(-0.01 \cdot x) \cdot [\sin x + 0.5 + 2 * \cos x]$. We compute these functions for 500 discrete values of x in the given interval, and then plot these functions.

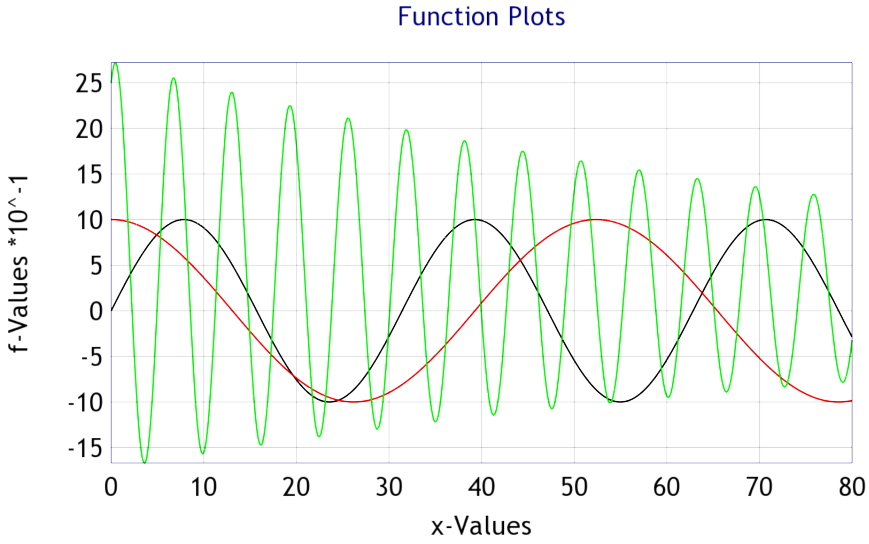

Figure 1: Graphics generated by *SLangTNG*

```
--[[
SLangTNG
Simple test example for mathematical function
(c) 2009 Christian Bucher, CMSD-VUT
--]]

   control.Interactive(true)

-- Set up an array for the x-values
   ar = tmath.Matrix(200)
   ar:SetLinearRows(0,80)
-- Apply the first function
   h=tmath.Sin(ar, 0.2)
   final=h*1
-- Same for second function and  and append to result
   h=tmath.Cos(ar, 0.12)
   final=final:AppendCols(h)
-- Ditto for third function
--   fun1=tmath.Exp(ar, -.01)
   fun1 = tmath.CWise(ar*(-.01), math.exp);
   fun2=tmath.Sin(ar)
   fun3=tmath.Cos(ar)
   fun=fun1-fun2+fun3*2
   print("fun", fun)
   final=final:AppendCols(fun)
   print("final", final)

-- Create graphics window and plot functions
   w = tnggraphics.TNGVisualize(.1, .1, .4, .4, "Welcome to SLangTNG!")
   w:SetLabels("Function Plots", "x-Values", "f-Values")
   w:Hold(true)
   w:Plot(ar, final)

   z=tmath.Matrix(1,10)
   z:SetLinearCols(10, 70)

   zz = tmath.Cos(z, 0.2)
   print("zz", zz)
   w:Plot(z, zz, -.01, 3) -- Symbol size < 0: plot only symbol
   w:Plot(z, tmath.Sin(z), .01, 1) -- Symbol size >= 0 plot line and symbol
   col=tmath.Matrix(4)
```

```
42    col[0] = 255; col[3] = 128;
43  --  w:Plot(tmath.Exp(ar, -.1), .01, 2, col);
44
45    u=tnggraphics.TNGVisualize(.5, .5, .4, .4, "Logarithmic plot")
46    u:SetLabels("Log Plots", "x-Values", "f-Values")
47    u:Frame(true)
48    u:Axes(true)
49    u:Logarithmic(false, true, false)
50    col[2] = 255; col[3] = 64;
51    t = tmath.Matrix(40)
52    t:SetLinearRows(0,80)
53    haha = tmath.Exp(t, -.1)
54    u:Plot(haha, .01, 2, col)
55    w:File("intro.pdf")
56
57  -- Send plots to CBDraw
58    tmath.CBDraw(ar, final, "final.cb")
59    tmath.CBDraw(ar, final, "final2.cb", 512, 384, "0 10 6 %g -3 3 7 %.2f 'The $x$-Axis
          ' 20 'The $y$ or $x^2$-Axis' 40")
```