Testing speed of fft in MatDeck and Python

In this example, we test the speed of fft function in Matdeck, and the speed of corresponding ffft function in Python. The variable nn is used to set the length of the vector which will be used to calculate fft. Here, we generate random signal using function normrandvec, with mean value equal to 0 and standard deviation equal to 1.

```
1  nn := 10003
2  x_in := normrandvec(0, 1, nn)
3  xx := vector create(nn, false, 0)
```

Next, we calculate fft using fft1 function. We measure the execution time using timenow() function. The function returns a complex vector X_in1 of the same size as argument vector, that is nn.

```
4  a1 := timenow()
5  X_in1 := fft1(x_in)
6  a2 := timenow()
```

a2 - a1 = 0.142 Time in seconds

size(X_in1) = 10003 Size of the complex vector

FFT in Python

Python code used to calculate fft is given below. Here, the same input random vector is used, by transferring MatDeck variable to Python. The function fft is part of scipy library which should be additionally installed. Therefore, it is necessary to install scipy, which can be done using the following command python -m pip install -U scipy.

```
7  b1 := timenow()
8  #py
9  from scipy.fftpack import fft
10 import time
11 start = time.time()
12 ss = fft(x_in)
13 end = time.time()
14 print (end - start)
15 ###
16 b2 := timenow()
```

We measure the time two times, for the hole Python block which is displayed in canvas below. The time needed to calculate fft in Python is displayed in system console as it is measured directly in Python.

b2 - b1 = 6.822