正弦波関数の自作(1)

演習15.　三角関数 $\sin x, \cos x, \tan x$ を作成せよ。

```
Option Explicit
Function my_sin(x As Double, nmax As Integer) As Double
'****************************************************************************
'**** 級数展開した際の各項の商が常にx^2のオーダーであることを利用する方法
'**** 参考文献 http://www.dfx.co.jp/dftalk/?p=8145
'****************************************************************************
Dim sum As Double
Dim t As Double
Dim pi As Double
Dim n As Double

pi = 3.14159265358979          '円周率πの定義:誤差10^(-14) オーダー
x = x - CInt(x / (2 * pi)) * 2 * pi  '範囲を-2π～2πにする
sum = x ' 級数の合計値
t = x  ' 初期項の値

For n = 1 To nmax
    t = t * -(x * x) / ((2 * n + 1) * (2 * n))  '第n項と第n-1項の商を第n-1項に掛ける
    sum = sum + t '
Next

my_sin = sum

End Function
```

デジタル・フロンティア-
Digital Frontier | DF TALK |
math(数学)関数を自作しよう！
https://dftalk.jp/?p=8145

※ 続く

正弦波関数の自作(2)

演習15.　三角関数 $\sin x, \cos x, \tan x$ を作成せよ。

※ 続き

```
Function my_cos(x As Double, nmax As Integer) As Double
'****************************************************************************
'**** 自作のsinを呼び出す方法
'**** 参考文献 http://www.dfx.co.jp/dftalk/?p=8145
'****************************************************************************
Dim pi As Double
pi = 3.14159265358979          '円周率πの定義:誤差10^(-14) オーダー

my_cos = my_sin(pi / 2 - x, nmax)

End Function
Function my_tan(x As Double, nmax As Integer) As Double
'****************************************************************************
'**** 自作のsinとcosを呼び出す方法
'**** 参考文献 http://www.dfx.co.jp/dftalk/?p=8145
'****************************************************************************
Dim pi As Double
pi = 3.14159265358979          '円周率πの定義:誤差10^(-14) オーダー

my_tan = my_sin(x, nmax) / my_cos(x, nmax)

End Function
```


正弦波関数の自作(計算原理1)

f(x)=sin x のマクローリン級数展開は

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!}$$

$$\text{又は} = \sum_{n=1}^{\infty} (-1)^{n-1} \frac{x^{2n-1}}{(2n-1)!}$$

プログラムでは無限級数を扱えないので、有限項で打ち切る必要がある。これが打切り誤差(truncation error) になる。

$$\sin x = \underbrace{+\frac{x}{1!}}_{t_1} \underbrace{-\frac{x^3}{3!}}_{t_2} \underbrace{+\frac{x^5}{5!}}_{t_3} \underbrace{-\frac{x^7}{7!}}_{t_4} + \cdots \underbrace{+(-1)^{n-1}\frac{x^{2n-1}}{(2n-1)!}}_{t_n} \underbrace{+(-1)^n \frac{x^{2n+1}}{(2n+1)!}}_{t_{n+1}}$$

$$= t_1 + t_2 + t_3 + t_4 + \cdots + t_n + t_{n+1}$$

ただし、t は term (項)の略である。

級数の総和を毎回計算してもよいが、四則演算の回数が増えて(計算に時間が掛かって応答が悪くなるため)効率的ではない。そこで、次のように四則演算を減らす工夫をする。

デジタル・フロンティア-Digital Frontier | DF TALK | math(数学)関数を自作しよう！ https://dftalk.jp/?p=8145

正弦波関数の自作(計算原理2)

各項の商を計算すると,

$$\frac{t_2}{t_1} = \frac{-\frac{x^3}{3!}}{+\frac{x}{1!}} = -\frac{1!x^2}{3!} = -\frac{x^2}{3\cdot 2} \longrightarrow t_2 = -\frac{x^2}{3\cdot 2} t_1$$

$$\frac{t_3}{t_2} = \frac{\frac{x^5}{5!}}{-\frac{x^3}{3!}} = -\frac{3!x^2}{5!} = -\frac{x^2}{5\cdot 4} \longrightarrow t_3 = -\frac{x^2}{5\cdot 4} t_2$$

$$\frac{t_4}{t_3} = \frac{-\frac{x^5}{7!}}{\frac{x^3}{5!}} = -\frac{5!x^2}{7!} = -\frac{x^2}{7\cdot 6} \longrightarrow t_4 = -\frac{x^2}{7\cdot 6} t_3$$

⋮

次の項 t_{n+1} は前の項 t_n の $-x^2/(2n+1)2n$ 倍になっている。

$$\frac{t_{n+1}}{t_n} = \frac{(-1)^n \frac{x^{2n+1}}{(2n+1)!}}{(-1)^{n-1} \frac{x^{2n-1}}{(2n-1)!}} = -\frac{(2n-1)!x^2}{(2n+1)!} = -\frac{x^2}{2n+1\cdot 2n} \longrightarrow \boxed{t_{n+1} = -\frac{x^2}{2n+1\cdot 2n} t_n}$$

デジタル・フロンティア-Digital Frontier | DF TALK | math(数学)関数を自作しよう！ https://dftalk.jp/?p=8145

正弦波関数の自作（計算原理3）

第n+1項目を求める手順は

$$t_1 = x \quad （初期値）$$

$$t_2 = -\frac{x^2}{3\cdot 2}t_1$$

$$t_3 = -\frac{x^2}{5\cdot 4}t_2$$

$$t_4 = -\frac{x^2}{7\cdot 6}t_3$$

$$\vdots$$

$$t_{n+1} = -\frac{x^2}{2n+1\cdot 2n}t_n$$

これをプログラミングで表すと、

$$\mathbf{t = t \times \left(\text{-}\frac{x^2}{(2n+1)\times(2n)} \right)}$$

右辺の t（初期値）が (−x²/(2n+1)2n) 倍されて左辺の t に代入される。この瞬間に、右辺に入っていた t は上書きされて次の計算の初期値になる。

一方、総和を求める式は

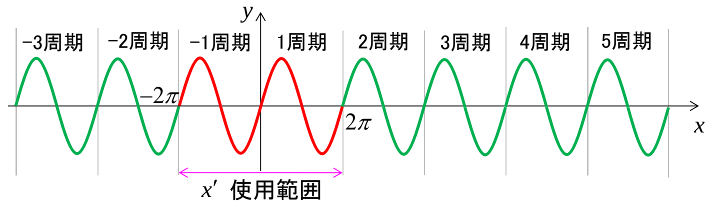
$$\sin x = t_1 + t_2 + t_3 + t_4 + \cdots + t_n + t_{n+1}$$

これをプログラミングで表すと、

$$\mathbf{sum = sum + t}$$

正弦波関数の自作（計算原理4）

所で、マクローリン級数展開は x=0 の近傍で関数を展開している。このため、x が原点から離れて大きな値になると誤差も大きくなる。そこで、三角関数の周期性を利用して、x>2π, x<−2π のような大きな x の値でも −2π<x<2π の範囲のxに置き換えて計算する。



$$x' = x - \mathrm{int}\left(\frac{x}{2\pi}\right)\times 2\pi$$

これをプログラミングで表すと、

$$\mathbf{x = x - int\left(\frac{x}{2\pi}\right)\times 2\pi}$$

右辺に入力されたxの周期（整数に丸める）を求めて、それを2π倍した値を引くことで、±1周期の中のxに置き換える。