any_PI について

香川高等専門学校 詫間キャンパス 現代視覚研究愛好会
情報工学科 5 年 T.K. (@AkihisaYoshii4)

まえがき

円周率というものは不思議だ．2 千年も前に発見しているのに，わからないことがとても多い．いつからだろう，私が円周率を求めて遊ぶ様になったのは．最初は C 言語で記述して CSV ファイルに保存し，Gnuplot や Excel でグラフ化していた．いろいろなアルゴリズムによって，収束の速さが異なるので，いろいろな結果ができた．そこから，桁数への挑戦に進んだ．C 言語で多倍長整数演算を行うライブラリを作り(現在はプログラムに間違いがあることを認識)，円周率を求めるためのテスト段階として，学校の演習用サーバで$2^{2^{32}}$の演算により，300GB 程度のファイルを作りだし，怒られた．それから，しばらくは特に何もなかったが，今回久しぶりに円周率で遊んでみようと思う．なお，今回のプログラムは時間の制約上，主に Integer 型と Double 型により実装しているので，細かい桁数は求められない．まあ，遊びか実験か自作の素材にでも使ってくれるとありがたい．今回のプログラムは完全に 1 人で 0 から作りだした，と言いたいところだが，実際には，実験で使用した関数を少し使いまわしている．まあ，0 から作るとしたら，半月程度はかかるのではないだろうか．多分，テストとかを入れると 1 ヶ月以上は余裕でかかると思う．そう考えると，プログラムというものは極力使いまわせる様にするべきだと思う．頑張れオブジェクト指向．せっかくだから，このプログラムを作った日誌を示す．

2017/12/04

さあ，最後の総文だ．何か作ろう．C 言語で多倍長整数演算でもリメイクするか．和と差さえ作れば，積と商は簡単に実装できるだろう．

自然数同士の和と差は実装したが，整数の範囲になると，どちらも怪しいな．日付からみても多分間に合わないだろう．

2017/12/05

円周率の経過グラフとか綺麗だろうな．自分なら適当に作って，後から多倍長演算対応できるように作れるだろう．とりあえず，今回はそれでいいか．

Wikipedia をみて，適当な級数を実装．

2017/12/06

管理がめんどくさいな．グラフに関する関数をクラス化して，ファイルも分割しよう．Window も．級数ももっと増やそう．けど，級数だけじゃ面白みにかけるな，積分も実装しよう．

2017/12/07

処理時間程度は分かるようにするか．積分ももっと追加しよう．1 個 Window 増やせたんだから，2 個も変わらんだろ．モンテカルロ法実装しよう．目盛りも実装しよう．やべえ，Readme 書いてない・

2017/12/08

目盛りは小数単位で表示できた方がおもしろいな．改良しよう．Readme めちゃ長い….

1. 概要

本プログラムは円周率をいろいろな式で求め，途中の値を逐次グラフで表示するソフトウェアである．手軽に式の追加が可能なので，教育用に，使用できるといえる．また，Microsoft 社による Visual Basic で開発されていることから，プログラムへの知識が低い人でも手軽に処理，GUI の改変が可能である．

2. 起動

以下の起動方法がある．

(ア) Visual Studio で"any_PI.sln"を実行し，F5 でデバッグモードとして，起動する．

(イ) フォルダを置き，"Release_any_PI.exe"をダブルクリックで実行する．

プログラムへの改変等を行う場合は，”(ア)”の方法を用いなければならない．

3. 使い方

以下の内容に従って，数値を入力する．

項目	内容	型	影響するグラフ
演算回数	実際に演算を行う回数	整数	any_PI，Monte_Carlo
Min x	表示するx軸の最小値	整数	any_PI
Max x	表示するx軸の最大値	整数	any_PI
Min y	表示するy軸の最小値	整数	any_PI
Max y	表示するy軸の最大値	整数	any_PI
点の直径	グラフに打つ dot の直径	整数	any_PI
x 目盛り幅	x軸の目盛り幅	整数 or 小数	any_PI
y 目盛り幅	y軸の目盛り幅	整数 or 小数	any_PI

その後，”INIT”ボタンを押下し，値を適応させた後に，”START/STOP”ボタンで処理が開始する．処理の途中に同ボタンを押下すると，処理が一時中断され再び押下すると，処理が続行される．

また，以下の内容に従って，値が表示される．

項目	内容
現在の演算回数	現在の演算回数
Math.PI	System.Math.PI の値
所要時間	処理を行っている実時間
各公式の”値”列	各公式のその時点での演算結果
各公式の”差分”列	演算結果と System.Math.PI の値との差
各公式の”Status”列	NaN エラー発生時に赤字で演算エラーを表示

any_PI グラフと各公式の”値”列の色は対応している．

4. 処理時間について

処理のスタートからストップまでを System.Diagnostics.Stopwatch を使って，計測している．INIT ボタンを押下すると，Reset メソッド，START/STOP ボタンを押下すると，Timer1 が動作しているときは Stop メソッド，Timer1 が停止しているときは Start メソッドが実行される．

5. 各ファイルの内容

(ア) Form1.vb(any_PI コントロールパネル)

各クラスの操作を行う．また，初期値の設定，値渡しなどを行う．

(イ) Show_Dot_Graph.vb(any_PI グラフ)

各クラスの演算結果を Form1.vb から受け取り，dot を打つ．

(ウ) Integral_Calc_PI.vb

積分を使い，円周率を求める演算を行う．

(エ) Sigma_Calc_PI.vb

級数を使い，円周率を求める演算を行う．

(オ) Monte_Carlo_Calc_PI.vb

モンテカルロ法を用いて，円周率を求める演算を行う．

(カ) Monte_Carlo_Circle_Form.vb(Monte_Carlo グラフ)

“(オ)”にて，生成された座標にあわせて，dot をグラフに打つ．

(キ) Web_Data_Array_PI.vb

未実装.

6. 実装した円周率の求め方

(ア) Integral_Calc_PI.vb

積分を用いる．本プログラムでは，主に以下のように積分を行う．

$$F(x) = \int_a^b f(x)dx$$

のとき，

$$dt = \frac{b-a}{dx}$$

とすると，

$$F(x) \approx \sum_{x=a}^{b-1} f(x)dt$$

となり，近似できる．このとき，a, bの値は Integral_Calc_PI.vb 内の Integral 構造体に初期値として，入力する．また，dtは演算回数nを用いて，

$$dt = \frac{b-a}{n}$$

と表す．

① ガウス積分(Gaussian Integral)

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$$

正規分布の正規化定数の計算などに用いられる．本プログラムでは，以下のようにして，結果を得る．また，∞による演算は行えない為，$\infty \approx 1.0 \times 10^3$として，演算を行った．

$$\pi = \left(\int_{-1.0\times10^3}^{1.0\times10^3} e^{-x^2} dx\right)^2$$

よって，円周率が求まる．

② 積分_1(内部の変数は seki_1)

$$\pi = 2\int_{-1}^{1} \sqrt{1-x^2}\, dx$$

③　積分_2(内部の変数は seki_2)

$$\pi = 4\int_0^1 \frac{1}{1+x^2}dx$$

(イ) Sigma_Calc_PI.vb

級数を用いる．束縛変数 n は 0 から，Form1.vb の TextBox1(演算回数)に入力した値まで，演算を行う．

①　ライプニッツの公式(Leibniz Formula)

$$\pi = 4\left(1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\cdots\right) = 4\sum_{n=0}^{\infty}\frac{(-1)^n}{2n+1}$$

マーダヴァ-ライプニッツ級数と呼ばれることもある．以下にフーリエ級数を用いた証明を示す．

方形波$f(x)$を

$$f(x) = \begin{cases} -1 & -\pi \le x < 0 \\ 1 & 0 \le x < \pi \end{cases}$$

と定義する．フーリエ係数a_nは奇関数なので，

$$a_n = 0$$

であり，b_nは，

$$\begin{aligned} b_n &= \frac{1}{\pi}\int_{-\pi}^{\pi} f(x) sinnx\, dx \\ &= \frac{1}{\pi}\left\{\int_{-\pi}^{0}(-1) sinnx\, dx + \int_0^{\pi} 1 \cdot sinnx\, dx\right\} \\ &= \begin{cases} \dfrac{4}{n\pi} & n = 2k+1 \\ 0 & n = 2k \end{cases} \quad (k \in \boldsymbol{N}) \end{aligned}$$

と求められる．したがって，方形波$f(x)$のフーリエ級数は

$$f(x) = \frac{4}{\pi}\left(sinx + \frac{1}{3}sin3x + \frac{1}{5}sin5x + \cdots\right)$$

とあり，$f(x)$は$x = \frac{\pi}{2}$において連続であるので，両辺に代入すると，

$$1 = \frac{4}{\pi}\left(1-\frac{1}{3}+\frac{1}{5}-\cdots\right)$$

と，表すことができる．これを整理すると，

$$\pi = 4\left(1-\frac{1}{3}+\frac{1}{5}-\cdots\right)$$

$$= 4\sum_{n=0}^{\infty}\frac{(-1)^n}{2n+1}$$

が求められる．

②　リーマンゼータ関数(Riemann zeta function)

$$\zeta(s) = \sum_{n=1}^{\infty}\frac{1}{n^s}$$

と定義されるとき，$s = 4$ならば，

$$\zeta(4) = \sum_{n=1}^{\infty} \frac{1}{n^4}$$
$$= \frac{\pi^4}{90}$$

である． 整理すると，

$$\pi = \left(90 \sum_{n=1}^{\infty} \frac{1}{n^4}\right)^{\frac{1}{4}}$$

である．

③ ガウス＝ルジャンドルのアルゴリズム(Gauss=Legendre algorithm)

以下の初期値を設定する．

$$a_0 = 1$$
$$b_0 = \frac{1}{\sqrt{2}}$$
$$t_0 = \frac{1}{4}$$
$$p_0 = 1$$

以上の初期値を用い，以下の反復計算を行う．

$$a_{n+1} = \frac{a_n + b_n}{2}$$
$$b_{n+1} = \sqrt{a_n b_n}$$
$$t_{n+1} = t_n - p_n(a_n - a_{n+1})^2$$
$$p_{n+1} = 2p_n$$

十分に反復を行い，以下の式で円周率を求める．

$$\pi \approx \frac{(a+b)^2}{4t}$$

④ Chudonovsky の公式

$$\frac{1}{\pi} = \frac{12}{\sqrt{C^3}} \sum_{k=0}^{\infty} \frac{(-1)^k (6k)!\,(A+Bk)}{(3k)!\,(k!)^3 C^{3k}}$$

ただし，

$$\begin{cases} A & = & 13591409 \\ B & = & 545140134 \\ C & = & 640320 \end{cases}$$

である．整理すると，

$$\pi = \frac{1}{\frac{12}{\sqrt{C^3}} \sum_{k=0}^{\infty} \frac{(-1)^k (6k)!\,(A+Bk)}{(3k)!\,(k!)^3 C^{3k}}}$$

である．また，本公式はいくつかの書き方が存在するが，本プログラムでは，示した通りの式を用いる．

(ウ) モンテカルロ法

積分でグラフを描きランダムに点を打つことで，その点が積分範囲内か否かの面積比で積分を行う．本プログラムにおいては，

$$y(x) = \sqrt{r^2 - x^2} \quad \begin{cases} 0 \le x \le 1 \\ 0 \le y \le 1 \\ r = 1 \end{cases}$$

…3-1

にて演算を行う．この場合，点(x_1, y_1)において，

$$\begin{cases} 0 \le x_1 \le x \\ 0 \le y_1 \le y(x_1) \end{cases}$$

のときに，点は積分範囲内に入ったといえる，ここで，積分範囲に入った点の個数をn点，全体に打った回数をmをすると，式 3-1 より，

$$n : m = \frac{1}{4}\pi r^2 : r^2$$

より，

$$\frac{1}{4}\pi m r^2 = n r^2$$

$$\pi = 4\frac{n}{m}$$

と求まる．また，この方法は式，積分範囲，面積比を変更することで，どのような積分演算も可能である．

7. 関数を追加実装する方法

(ア) Form1.vb

① デザインに，出力用の TextBox を 3 つ追加する．

② Private Sub show_pi_value()へ以下を記述する．
show_pi_data(値を表示する TextBox, System.Math.PI との差分を表示する TextBox, 現在の演算状況を表示する TextBox, 値の色,グラフに打つ dot の色，各クラスの値を返すメソッド)

(イ) 積分演算の場合…Integral_Calc_PI.vb

① ファイル前半部分に Private 変数として，それぞれ演算に必要な変数を用意する．

② reset メソッドにて，各変数の初期化を行う．

③ Private Sub calc_処理名(n As Integer)を作成し，1 ルーチンごとに演算を行う．

④ Private Function get_calc_処理名()を作成し，πの値を返す．

⑤ update メソッドに”③”で作成した関数を追加する．

(ウ) 級数演算の場合…Sigma_Calc_PI.vb
“(イ)”と同様

(エ) モンテカルロ法の場合…Monte_Carlo_Calc_PI.vb
“(イ)”と同様

8. コード記法

以下に Visual Basic の記法について，日本語の意味とサンプルコードを示す．また，この記述以外にも様々な記法，構造があるが，今回は基礎部分のみを示す．

(ア) 変数宣言

```
Dim 変数 As 型
Dim x As Integer
```

(イ) For 文

For　変数=初期値　To　終了値　Step　増分値 処理 Next　変数	For i=0 To 10 Step 1 処理 Next i

(ウ) IF 文

IF　条件　Then
処理
ElseIf　条件　Then
処理
Else
処理
End IF
	IF　x>10 Then
	処理
	ElseIf x=10 Then
	処理
	Else
	処理
	End IF

(エ) Sub プロシージャ

Private Sub Sub 名(引数) 処理 End Sub	Private Sub Sub_Test (x As Integer) 処理 End Sub

(オ) Function プロシージャ

Private Function Function 名(引数) 処理 Return　変数 End Function	Private Function Func_Test(x As Integer) 処理 Return x End Function

(カ) 構造体

Private Structure　構造体名 要素(Dim ~) 要素(Dim ~) End Structure	Private Structure point Dim x As Integer Dim y As Integer End Structure

(キ) 参照渡し

① Sub プロシージャで受けとるとき

Private Sub Sub 名(ByRef　変数名) 処理 End Sub	Private Sub Sub_Addr (ByRef　p As Integer) 処理 End Sub

② Sub プロシージャに変数を渡す時

Sub 名(変数)

Sub_Addr(x)

③ Sub プロシージャに配列を渡す時

Sub 名(配列名括弧なし)

Sub_Addr(data)

9. 開発環境

(ア) Microsoft Windows 10 EnterPrice

Microsoft Visual Studio Community 2017

Version 15.4.1

VisualStudio.15.Release/15.4.1+27004.2005

Microsoft .NET Framework

Version 4.7.02046

(イ) Microsoft Windows 8.1 Embedded Industry Pro

Microsoft Visual Studio Community 2017

Version 15.4.1

VisualStudio.15.Release/15.4.1+27004.2005

Microsoft .NET Framework

Version 4.7.02053

参考文献

・ 円周率 – Wikipedia(https://ja.wikipedia.org/wiki/円周率) 2017/12/08
・ 円周率の歴史 – Wikipedia(https://ja.wikipedia.org/wiki/円周率の歴史) 2017/12/08
・ ガウス積分 – Wikipedia(https://ja.wikipedia.org/wiki/ガウス積分) 2017/12/08
・ リーマンゼータ関数 - Wikipedia(https://ja.wikipedia.org/wiki/リーマンゼータ関数) 2017/12/08
・ Chudnovsky の公式を用いた円周率の計算用メモ – Qiita (https://qiita.com/peria/items/c02ef9fc18fb0362fb89) 2017/12/08

第 1 版 2017/12/08

香川高等専門学校 詫間キャンパス 現代視覚研究愛好会
情報工学科 5 年 T.K. (@AkihisaYoshii4)