情報処理 2 第 2 回

十進BASIC (2) 繰り返しのある計算

桂田 祐史（かつらだ まさし）

2007 年 4 月 24 日

この授業用の WWW ページは `http://www.math.meiji.ac.jp/~mk/syori2-2007/`

今日は、コンピューターによる計算で最も大事な要素であると私が信じている「繰り返しのある計算」のプログラミングを取り上げます。同様のことを C 言語のプログラムでやった人も多いはずですが、気持ちを新たに「これくらいは自力で出来るようになってやる」と考えてチャレンジしてください。

1 連絡事項

- BASIC のヘルプがうまく表示されない問題ですが、とりあえず旧版の BASIC.HLP[1] を保存してクリックすると少々不便ながらも読めると思います。なお、これはあくまでも暫定措置で、この問題はきちんと解決する予定です。
- プログラムのファイルをダブル・クリックしたら十進 BASIC を起動するようにしたいという人へ: そのうちきちんと説明しますが、とりあえずヒントを出しておくと、実行を抑制するために `/NR` というオプションを指定するのがミソです。

2 FOR 〜 NEXT による繰り返し

コンピューターが最も能力を発揮するのは、**繰り返し計算**をしているときだと私は信じています。人間が書く短いプログラムで大規模な計算をするのは、プログラムのどこかで繰り返し処理があるからでしょう。

ここでは数学の問題を繰り返し処理を用いて解くプログラムを考えます。すべてとはいいませんが、かなりの部分は**漸化式**による計算になります。

コンピューターに繰り返し処理をさせるには、“**再帰的手続き**” などもありますが、ここではポピュラーな FOR 〜 NEXT という繰り返し構文を使ってみます。

[1] `http://www.math.meiji.ac.jp/~mk/syori2-2007/mo/BASIC.HLP`

2.1 FOR NEXT 構文の復習

十進 BASIC には繰り返しを記述するための命令が色々ありますが、多分 FOR NEXT 構文が一番よく使われるでしょう。

nenbutsu.bas

```
REM 単純な繰り返し
FOR n=1 to 100
  print "南無阿弥陀仏"
NEXT n
END
```

2.2 簡単な漸化式

例題 1. 定数 r が与えられたとき、漸化式

$$a_1 = 1, \quad a_n = ra_{n-1} \quad (n \geq 2)$$

で定義される数列 $\{a_n\}_{n\in\mathbf{N}}$ の最初の 100 項を求めよ。— 要するに等比数列 $a_n = r^{n-1}$ ですが、この仕事を要求するには、例えば次のようなプログラムを書けば OK です。

配列を使うバージョン touhi1.bas

```
REM 等比数列 (配列を使うバージョン)
DIM A(100)
INPUT PROMPT "r=": r
A(1)=1
FOR n=2 to 100
  A(n)=r*A(n-1)
  print n;A(n)
NEXT n
END
```

演算精度を上げておくべきかもしれません。これは (1000 桁演算モード) ボタンを押しても実現できますが、プログラムの先頭部分に例えば

```
OPTION ARITHMETIC DECIMAL_HIGH
```

と書いてもよいでしょう。

実は一度 A(n-1) として使われた後はもう使われなくなるので、次のようなプログラムで済ませることが出来ます。

配列を使わないバージョン touhi2.bas

```
REM 等比数列 (配列を使わないバージョン)
INPUT PROMPT "r=": r
A=1
FOR n=2 to 100
  A=r*A
  print n;A
NEXT n
END
```

2.3 課題 2A

フィボナッチ数列は、漸化式

$$a_1 = 1, \quad a_2 = 1, \quad a_n = a_{n-1} + a_{n-2} \quad (n \geq 3)$$

で定義されるが、この最初の 100 項を求めよ。配列を使ったプログラムを書いて、その結果と一緒にレポートしなさい (必修)。これは出席がわりで今日中にレポートを送ること。

もしできれば配列を使わないプログラムも作成しなさい。

レポートの提出の仕方について、もう一度だけ注意しておくと (以後繰り返さない)、宛先は `syori2@math.meiji.ac.jp`, Subject (表題) は「情報処理 2 課題 2A」, 本文先頭に学年・組・番号・氏名を明記。

例えばこんなふうにして下さい

```
桂田先生

２年１６組９９番 数学大郎です。

情報処理２の課題２のレポートです。
プログラム kadai2.BAS と実行結果 kadai2.TXT は添付してあります。

フィボナッチ数列の第 100 項は ○○○○ でした。
```

2.4 和 $\sum$ の計算

一見漸化式と関係なさそうでも、漸化式を用いて計算すると便利というものが結構あります。例えば数列 $\{a_j\}$ の和 $s = \displaystyle\sum_{j=1}^{n} a_j$ は、部分和

$$s_j := \sum_{k=1}^{j} a_k$$

を導入すると、数列 $\{s_j\}$ の第 n 項である、すなわち $s = s_n$ ですが、$\{s_j\}$ は

$$s_1 = a_1, \quad s_j = s_{j-1} + a_j \quad (j \geq 2)$$

あるいは

$$s_0 = 0, \quad s_j = s_{j-1} + a_j \quad (j \geq 1)$$

という漸化式で定義することができます。例えば

```
s=0
for j=1 to n
  s=s+(a_j を計算する式)
next j
print s
```

のようなコード[2]で s_n が計算できます。

2.5 課題 2B

自然数 n が与えられたときに

$$s_n := \sum_{j=1}^{n} \frac{1}{j}, \quad t_n := \sum_{j=1}^{n} \frac{1}{j^2}$$

を計算するプログラムを書き、$n = 1, 10, 100, 1000, \ldots$ のとき[3]、値がどうなるか調べ (記録を取ること)、説明しなさい。なお、$n \to \infty$ のとき s_n, t_n がどうなるか、実はよく知られたことです (収束・発散については基礎数学 4 で学んだはず)。それを知っているのならば、そのことを踏まえて結果を説明して下さい。締め切りは 5 月 7 日 (火曜) とします。

工夫のヒント: 工夫すると、1 つのプログラムで複数の n の値に対する s_n, t_n の値を一気に計算することができます。できれば、そういうプログラムを作ってみて下さい。

2.6 課題 2C1

これはレポートを提出するかどうか任意 (余裕がある人向けの「挑戦課題」)。締め切りはこの講義の最終回まで。

半径 1/2 の円の円周は円周率 π に等しい ($r = 1/2$ なので $2\pi r = 2\pi \cdot 1/2 = \pi$)。有名なシラクサのアルキメデス (BC287?〜BC212) は、内接正 n 角形の周の長さ p_n と、外接正 n 角形の周の長さ q_n を考え、

$$p_n < \pi < q_n, \quad \lim_{n\to\infty} p_n = \lim_{n\to\infty} q_n = \pi$$

[2]コンピューター・プログラム (特にその断片) のことをコードと言うことがあります。

[3]n が大きくなると計算に時間がかかります。概算でよければ精度は無暗に高くしないことを勧めます。十進 BASIC の場合は 1000 桁演算モードでない、普通の演算モードで計算すると速いでしょう。どれくらいスピードに差が出るでしょうか? また有理数演算モードでは?色々試してみてください。

をもとにして、実際に $n = 6, 12, 24, 48, 96$ に対して p_n, q_n を計算して π の評価を得た。

アルキメデスは $\{p_n\}$, $\{q_n\}$ について成り立つ漸化式 (?)

$$q_{2n} = \frac{2p_nq_n}{p_n+q_n}, \quad p_{2n} = \sqrt{p_nq_{2n}}$$

を用いて計算したという (現代の数学を使えば $p_n = n\sin(\pi/n)$, $q_n = n\tan(\pi/n)$ と表される。これは sin, tan の計算には直接は役立たないが、漸化式が成立することの確認には役立つかもしれない。)。

$n = 3\cdot 2^k$ の場合、すなわち内接・外接正 $3\cdot 2^k$ 角形の周長をそれぞれ P_k, Q_k とおく:

$$P_k := p_{3\cdot 2^k}, \quad Q_k := q_{3\cdot 2^k}.$$

このとき $\{P_k\}$, $\{Q_k\}$ について成り立つ漸化式を導き、

$$P_1 = p_3 = 3, \quad Q_1 = q_3 = 2\sqrt{3}$$

と合わせて用いて $\{P_k\}$, $\{Q_k\}$ を計算するプログラムを作成し、以下の (1), (2) に答えよ。

(1) 正 96 角形 ($k = 5$ の場合) を利用すると、π のどのような評価が得られるか。

(2) ネットで検索すると、内接正多角形の周長で円周率の近似値を求めた話が色々見つかる (円周率マニアは多い)。それらの話をいくつか選んで確認せよ (正 n 角形の周の長さを用いて小数点以下〇〇桁の値を求めたとある場合、本当にそれができるか — ときどきウソが書いてある)。

2.7 課題 2C2

これもレポートを提出するかどうか任意。締め切りはこの講義の最終回まで。

正数 a, b に対して、対数 $\log_a b$ の近似値を以下の手順で計算することができる。

$\log_a b$ の近似値の求め方 (by Briggs)

数列 $\{a_n\}$, $\{b_n\}$ を

$$\begin{aligned} &a_0 := a, \quad b_0 := b, \\ &a_{n+1} := \sqrt{a_n}, \quad b_{n+1} := \sqrt{b_n} \quad (n = 0, 1, 2, \dots) \end{aligned}$$

で定め、十分大きな n に対して

$$\frac{b_n - 1}{a_n - 1}$$

を $\log_a b$ の近似値に採用する。

Henry Briggs (1561–1630) はこの方法で歴史上初めての常用対数表を作成した (彼は $n = 54$ を採用したという、常用対数でなければ John Napier (1550–1617) が作成したもの (1614) が最初である)。

十進 BASIC では $\sqrt{a}$ は `SQR(A)` で計算できる。

以下の (1), (2) に答えよ。

(1) Briggs の方法で $\log_{10} 2$ を求め、

$$\log_{10} 2 = 0.30102\,99956\,63981\,19521\,37388\,94724\,49302\,67681\ldots$$

と比較せよ (精度はどの程度か)。

(2) なぜこの方法で log が計算できるか説明せよ (Briggs は微積分のない時代の人だが、微積分を使って説明しても構わない)。