

## GKS入門(5)

橋倉, 聡

九州大学大型計算機センターシステム管理掛

浦川, 伸治

九州大学大型計算機センターシステム運用掛

藤井, 龍磨

九州大学大型計算機センターシステム管理掛

河津, 秀利

九州大学大型計算機センターシステム管理掛

<https://doi.org/10.15017/1474996>

---

出版情報 : 九州大学大型計算機センター広報. 23 (5), pp.491-504, 1990-09-25. 九州大学大型計算機センター

バージョン :

権利関係 :



## G K S 入門 ( 5 )

橋倉 聡\*, 浦川 伸治\*\*, 藤井 龍磨\*, 河津 秀利\*

## 1. はじめに

G K S 入門シリーズも今回が最終回である。前回の入門(4)<sup>[4]</sup>では、K S T フォント、複数のワークステーション(G K S で用いる図形装置)の動作、セグメントについて説明した。K S T フォントは、出力する文字の高さや幅を自由に設定できるフォントである。セグメントは、折れ線や矩形などの出力基本要素をひとつの集まりとするものである。今回はメタファイルの使用法について説明する。

## 2. メタファイル

メタファイルとは、G K S で作成した図形や利用者が作成した任意のデータ(整数、実数、文字など。以後、利用者データと呼ぶことにする)を格納するファイルである。メタファイルに格納された図形はG K S により再生することができ、他の図形と組み合わせて新しい図形を作成することもできる。また、メタファイルに格納された利用者データはG K S により読み出して用いることができる。

## 2. 1 メタファイルのワークステーション種別

メタファイルは、一種のワークステーションとして扱われる。このため、ワークステーション制御機能(表示面消去など)によって制御される。メタファイルのワークステーション種別には以下のものがある。

## ・ M O ( M E T A F I L E O U T P U T )

メタファイルに図形、または任意のデータを格納する場合のワークステーション種別である。この種別には、装置座標の大きさ、方向の異なるものが用意されている。これを本センタールでは、

**M O # # @**

のように表現している。##は装置座標の大きさを示し、A 4、B 4、B 5のいずれかである。しかし、O P R や C L P の装置座標とは大きさがかなり異なっている。@は方向でL (ランドスケープ:横長)、P (ポートレート:縦長)のどちらかである。

## ・ M I ( M E T A F I L E I N P U T )

メタファイルに既に格納されている図形やデータを再生する場合のワークステーション種別である。この種別は、すべての既存のメタファイルを扱うことができる。

なお、ワークステーション記述表(使用するワークステーションの能力に関する情報を管理している。具体的には線種や表示面のサイズ等である)の問い合わせ機能はほとんど使用できない。たとえば、入門(2)<sup>[2]</sup>でG Q D S P という、使用するワークステーションの表示面の大きさを問い合わせるサブルーチンについて触れたが、ワークステーションがメタファイルの場合はエラー

平成2年7月27日受理

\* 九州大学大型計算機センターシステム管理掛

\*\* " システム運用掛



### 3. 2 メタファイルに図形を格納するプログラム例

グラフを出力してメタファイルに格納するプログラムである。このプログラムによってメタファイルに格納される図形を図3. 1に示す。

(プログラム例)

```

0001     REAL X(401), Y(401), FX(4), FY(4), UPPER(2), LOWER(2), LEFT(2), RIGHT(2),
0002     +     CHX(2), CHY(2), RAD, XSTEP, YSTEP
0003     CHARACTER*5 XCHAR, YCHAR
0004     PARAMETER (RAD=3.1415/180.0)
0005 C
0006     DO 1 I=1, 401
0007         X(I) = FLOAT(I-1)
0008         Y(I) = SIN(X(I)*RAD)/1.5
0009     1 CONTINUE
0010 C
0011     FX(1) = 0.0
0012     FX(2) = 400.0
0013     FX(3) = FX(2)
0014     FX(4) = FX(1)
0015     FY(1) = -1.0
0016     FY(2) = FY(1)
0017     FY(3) = 1.0
0018     FY(4) = FY(3)
0019 C
0020     XSTEP = (FX(2)-FX(1))/10.0
0021     YSTEP = (FY(3)-FY(1))/10.0
0022     UPPER(1) = FY(3)+YSTEP/10.0
0023     UPPER(2) = FY(3)-YSTEP/10.0
0024     LOWER(1) = FY(1)+YSTEP/10.0
0025     LOWER(2) = FY(1)-YSTEP/10.0
0026     LEFT(1) = FX(1)+XSTEP/10.0
0027     LEFT(2) = FX(1)-XSTEP/10.0
0028     RIGHT(1) = FX(2)+XSTEP/10.0
0029     RIGHT(2) = FX(2)-XSTEP/10.0
0030 C
0031     CALL GOPKS (0,1)
0032     CALL GOPWK (1,1,1)
0033     CALL GACWK (1)
0034 C
0035     CALL GSWN (1,-40.0,440.0,-1.2,1.2)
0036     CALL GSVL (1,0.0,1.0,0.0,1.0)
0037     CALL GSELNT (1)
0038 C

```

```

0039      CALL GSLN  (1)
0040      CALL GPL   (401, X, Y)
0041      CALL GFA   (4, FX, FY)
0042      CALL GSTXFP (-100, 2)
0043      CALL GSCHH (YSTEP/7.5)
0044      CALL GSCHXP (XSTEP/YSTEP)
0045      DO 2 I=0, 10
0046          CHX(1) = FX(1)+XSTEP*FLOAT(1)
0047          CHX(2) = CHX(1)
0048          CALL GPL (2, CHX, UPPER)
0049          CALL GPL (2, CHX, LOWER)
0050          CHY(1) = FY(1)+YSTEP*FLOAT(1)
0051          CHY(2) = CHY(1)
0052          CALL GPL (2, LEFT, CHY)
0053          CALL GPL (2, RIGHT, CHY)
0054          WRITE (XCHAR, '(F5.1)') CHX(1)
0055          WRITE (YCHAR, '(F5.1)') CHY(1)
0056          CALL GSTXAL (2, 1)
0057          CALL GTX (CHX(1), LOWER(2), XCHAR)
0058          CALL GSTXAL (3, 3)
0059          CALL GTX (LEFT(2), CHY(1), YCHAR)
0060      2 CONTINUE
0061 C
0062      CALL GDAWK (1)
0063      CALL GCLWK (1)
0064      CALL GCLKS
0065 C
0066      STOP
0067      END

```

(プログラムの説明)

0001 ~ 0004 必要な変数・配列の宣言。  
 0006 ~ 0009 曲線の座標値の計算。  
 0011 ~ 0018 座標軸を描くための値を設定。  
 0020 ~ 0029 目盛り・目盛り数値を描くための値を設定。  
 0031 ~ 0033 G K S の開始, およびワークステーションを出力可能状態にする。  
 0035 ~ 0037 座標変換. ここでは, 正規化変換のみを行い, ワークステーション変換は G K S  
 に任せている。  
 0039 ~ 0040 曲線の出力。  
 0041 座標軸の出力。  
 0042 ~ 0044 文字のフォント, 表示精度, 字高, 文字幅比の設定。  
 0045 ~ 0060 目盛り・目盛り数値の出力。  
 0062 ~ 0064 ワークステーションを使用不可能状態にし, G K S を終了する。

(プログラムの実行)

META1.DATA というメタファイルに図形を格納する。ワークステーション種別は MOA4L とする。  
 READY  
 GKS85 プログラム名 WS(MOA4L(META1.DATA))

(メタファイルに格納される図形)

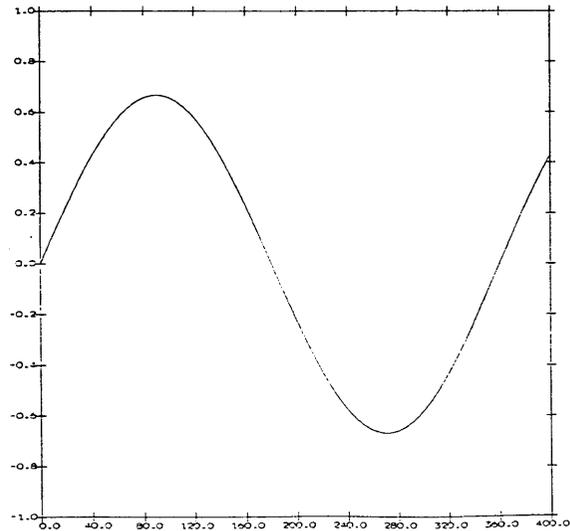


図 3. 1

### 3. 3 利用者データのメタファイルへの格納

メタファイルには、利用者データを格納することもできる。使用するサブルーチンを表 3. 1 に示す。

表 3. 1 利用者データ格納時に使用するサブルーチン

サブルーチン名	機能
G P R E C	利用者データを格納するデータレコードの作成
G W I T M	利用者データのメタファイルへの出力

利用者データをメタファイルに格納するためには、まず、データを格納したデータレコード（整数データ、実数データ、文字データを一つの領域にまとめたもの）を作成する必要がある。これは入門（3）<sup>[3]</sup>で説明したサブルーチン G P R E C を使用する。サブルーチンの詳細については入門（3）を参照して頂きたいが、データレコードの構造について詳しく触れなかったのでここで説明したい。

データレコードの長さは 1 レコード当たり 80 バイトである。データは 16 進数で格納されており、図 3. 2 のような構造になっている。

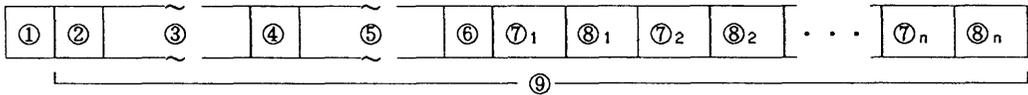


図 3. 2

- ① : ⑨の長さ (大きさは4バイトで, 格納される値の単位はバイト) .
- ② : 整数データの個数 (大きさは4バイト) .
- ③ : 整数データ (大きさは4バイト×②の値, すなわち②が0ならばこの部分はない) .
- ④ : 実数データの個数 (大きさは4バイト) .
- ⑤ : 実数データ (大きさは4バイト×④の値, すなわち④が0ならばこの部分はない) .
- ⑥ : 文字データの個数 (大きさは4バイト) . ここでは n が入っているとす.
- ⑦<sub>1</sub> : ⑧<sub>1</sub>の文字データ<sub>1</sub>の長さ (大きさは4バイトで, 格納される値の単位はバイト. ⑥が0ならばこの部分はない) .
- ⑧<sub>1</sub> : 文字データ<sub>1</sub> (大きさは⑦<sub>1</sub>の値. ⑥が0ならばこの部分はない) .
- ⑦<sub>2</sub> : ⑧<sub>2</sub>の文字データ<sub>2</sub>の長さ (大きさは4バイトで, 格納される値の単位はバイト. ⑥が0ならばこの部分はない) .
- ⑧<sub>2</sub> : 文字データ<sub>2</sub> (大きさは⑦<sub>2</sub>の値. ⑥が0ならばこの部分はない) .
- :
- :
- ⑦<sub>n</sub> : ⑧<sub>n</sub>の文字データ<sub>n</sub>の長さ (大きさは4バイトで, 格納される値の単位はバイト. ⑥が0ならばこの部分はない) .
- ⑧<sub>n</sub> : 文字データ<sub>n</sub> (大きさは⑦<sub>n</sub>の値. ⑥が0ならばこの部分はない) .

データレコードの次元数はこの図 3. 2 を参考に設定するとよい.

データレコードを作成した後, これを元にしてデータをメタファイルに格納するために G W I T M というサブルーチンを使用する. 以下に呼び出し形式を示す.

```
CALL GWITM(WKID, TYPE, IDRL, LDR, DATREC)
```

引数 WKID はデータを出力するワークステーション名 (メタファイル) を指定する. 引数 TYPE は項目種で 101 以上の値を指定する. 引数 IDRL は利用者データのデータ長をバイト数で指定する. この値は引数 DATREC (データレコード) の有効データ長で, 図 3. 2 の①の値 (すなわち⑨の長さ) である. 具体的な設定方法は,

$$IDRL = 4 + \text{整数データの数} \times 4 + 4 + \text{実数データの数} \times 4 + 4 \left( + 4 + \text{文字データ}_1 \text{の長さ} + 4 + \text{文字データ}_2 \text{の長さ} + \dots + 4 + \text{文字データ}_n \text{の長さ} \right)$$

のようになる. 括弧で囲んだ部分は, 文字データをまったく格納しないのならば必要ない. 引数 LDR は引数 DATREC の次元数を指定する. 以上の引数はすべて4バイトの整数型である. 引数 DATREC はデータの格納されたデータレコードを指定する. この引数は長さ80バイトの文字型の1次元配列である.

### 3. 4 メタファイルに利用者データを格納するプログラム例

計算した値をメタファイルへ格納するプログラム例である.

## (プログラム例)

```

0001      REAL      X(401), Y(401), RAD
0002      INTEGER   ERR1, ERR2, LDR1, LDR2, IDMMY1, IDMMY2
0003      CHARACTER DATREC(21)*80, CDMMY
0004      PARAMETER (RAD=3.1415/180.0)
0005 C
0006      DO 1 I=1, 401
0007          X(I) = FLOAT(I-1)
0008          Y(I) = COS(X(I)*RAD)/2.0
0009      1 CONTINUE
0010 C
0011      CALL GOPKS (0, 1)
0012      CALL GOPWK (1, 1, 1)
0013      CALL GACWK (1)
0014 C
0015      CALL GPREC (0, IDMMY1, 401, X, 0, IDMMY2, CDMMY, 21, ERR1, LDR1, DATREC)
0016      CALL GWITM (1, 101, 4+4+401*4+4, 21, DATREC)
0017      CALL GPREC (0, IDMMY1, 401, Y, 0, IDMMY2, CDMMY, 21, ERR2, LDR2, DATREC)
0018      CALL GWITM (1, 102, 4+4+401*4+4, 21, DATREC)
0019 C
0020      CALL GDAWK (1)
0021      CALL GCLWK (1)
0022      CALL GCLKS
0023 C
0024      STOP
0025      END

```

## (プログラムの説明)

0001 ~ 0004 必要な配列・変数の宣言。配列 DATREC はデータレコードのことで、配列 X と Y がそれぞれ 401 の要素を持つから、 $(4+4+401*4+4)/80$  で 21 必要である。

0006 ~ 0009 値の計算。

0011 ~ 0013 G K S の開始、およびワークステーションを出力可能状態にする。

0015 配列 X の値をデータレコードに格納する。

0016 配列 X の値を格納しているデータレコードを用いてメタファイルに配列 X の値を格納する。項目種は 101 としている。

0017 配列 Y の値をデータレコードに格納する。

0018 配列 Y の値を格納しているデータレコードを用いてメタファイルに配列 Y の値を格納する。項目種は 102 としている。

0020 ~ 0022 ワークステーションを使用不可能状態にし、G K S を終了する。

## (プログラムの実行)

META2.DATA というメタファイルに利用者データを格納する。ワークステーション種別は MOA4L とする。

READY

GKS85 プログラム名 WS(MOA4L(META2.DATA))

### 3. 5 図形の再生

メタファイルに格納された図形を再生する場合に用いるサブルーチンを表 3. 2 に示す。

表 3. 2 図形再生時に使用するサブルーチン

サブルーチン名	機能
GGTITM	項目種・項目長の読み出し
GRDITM	項目データの読み出しとカレントポイントの更新
GIITM	項目種と項目データの翻訳

メタファイルに格納された図形（利用者データも同じであるが）は、項目種により管理されている（2. 2 参照）。メタファイルから読み出しを行うには、まず、メタファイルの中の項目種と項目長を読む。これを行うのはサブルーチンGGTITMである。以下に呼び出し形式を示す。

CALL GGTITM(WKID, ITEM, LENGTH)

引数 WKID はワークステーション名（メタファイル）を指定する。引数 ITEM はGKSが項目種を通知する領域である。引数 LENGTH はGKSが項目長を通知する領域である。いずれの引数も4バイトの整数型である。このサブルーチンでは、項目種、項目長を読んでプログラムに通知するだけである。従ってカレントポイントは更新しない。

項目データの読み出し、カレントポイントの更新はサブルーチンGRDITMによって行う。以下に呼び出し形式を示す。

CALL GRDITM(WKID, MIDRL, MLDR, DATREC)

引数 WKID はワークステーション名（メタファイル）を指定する。引数 MIDRL は項目長を指定する。このサブルーチンはサブルーチンGGTITMと併せて使用するので、実際にはサブルーチンGGTITMにより通知された項目長（引数 LENGTH）を指定すればよい。引数 MLDR は引数 DATREC の次元数を指定する。以上の引数はすべて4バイトの整数型である。引数 DATREC は項目データを格納するデータレコードを指定する。この引数は長さ80バイトの文字型の1次元配列である。

メタファイルに格納された図形を再生するには、もう一つサブルーチンが必要である。利用者データはメタファイルに格納されてもデータそのものは加工されないが、図形は完全に加工されているので、翻訳作業を行う。翻訳するサブルーチンはGIITMである。以下に呼び出し形式を示す。

CALL GIITM(TYPE, IDRL, LDR, DATREC)

引数 TYPE は翻訳する項目種を指定する。このサブルーチンはサブルーチンGGTITM、GRDITMと併せて使用するので、サブルーチンGGTITMにより通知される項目種（引数 ITEM）を指定すればよい。引数 IDRL は項目長を指定する。これもサブルーチンGGTITMにより通知される項目長（引数 LENGTH）を指定すればよい。引数 LDR は引数 DATREC の次元数を指定する。以上の引数はすべて4バイトの整数型である。引数 DATREC はサブルーチンGRDITMによりデータの格納されたデータレコードを指定する。この引数は長さ80バイトの文字型の1次元配列である。

## 3. 6 図形の再生を行うプログラム例

3. 2でメタファイルに格納した図形を再生し、新たな曲線を加えて他のワークステーションに出力するプログラム例である。このプログラムの出力結果を図3. 3に示す。

(プログラム例)

```

0001      REAL      X(401), Y(401), RAD
0002      INTEGER   ITEM, LENGTH
0003      CHARACTER DATREC(21)*80
0004      PARAMETER (RAD=3.1415/180.0)
0005 C
0006      DO 1 I=1, 401
0007          X(I) = FLOAT(I-1)
0008          Y(I) = COS(X(I)*RAD)/2.0
0009      1 CONTINUE
0010 C
0011      CALL GOPKS (0, 1)
0012      CALL GOPWK (2, 2, 2)
0013      CALL GOPWK (1, 1, 1)
0014      CALL GACWK (2)
0015 C
0016      2 CONTINUE
0017      CALL GGTITM (1, ITEM, LENGTH)
0018      IF (ITEM.EQ.0) GOTO 3
0019      CALL GRDITM (1, LENGTH, 21, DATREC)
0020      CALL GIITM (ITEM, LENGTH, 21, DATREC)
0021      GOTO 2
0022 C
0023      3 CONTINUE
0024      CALL GSWN (1, -40.0, 440.0, -1.2, 1.2)
0025      CALL GSVF (1, 0.0, 1.0, 0.0, 1.0)
0026      CALL GSELNT (1)
0027 C
0028      CALL GSLN (2)
0029      CALL GPL (401, X, Y)
0030 C
0031      CALL GDAWK (2)
0032      CALL GCLWK (1)
0033      CALL GCLWK (2)
0034      CALL GCLKS
0035 C
0036      STOP
0037      END

```

(プログラムの説明)

- 0001 ~ 0004 必要な変数・配列の宣言.
- 0006 ~ 0009 付け加える曲線の座標値の計算.
- 0011 ~ 0014 GKSの開始, およびワークステーション 1 を使用可能状態, ワークステーション 2 を出力可能状態にする.
- 0016 ~ 0018 ワークステーション 1 (メタファイルのMI 種別) から項目種と項目長を読む. 項目種が 0 ならば, データの終了と見なし, 0023 へ分岐する.
- 0019 ~ 0021 データレコードにメタファイルに格納されているデータを格納し, これを元に翻訳 (図形の再生) する. その後, 0016 へ分岐し, 項目種と項目長を読む.
- 0024 ~ 0026 座標変換. ここでは, 正規化変換のみを行い, ワークステーション変換はGKS に任せている.
- 0028 ~ 0029 曲線をメタファイルから再生した図形に付け加える.
- 0031 ~ 0034 ワークステーションを使用不可能状態にし, GKSを終了する.

(プログラムの実行)

実行結果をテクトロ端末 (T4014) に表示する.

READY

GKS85 プログラム名 WS('MI(META1.DATA), 2/2:T4014')

(プログラムの出力結果)

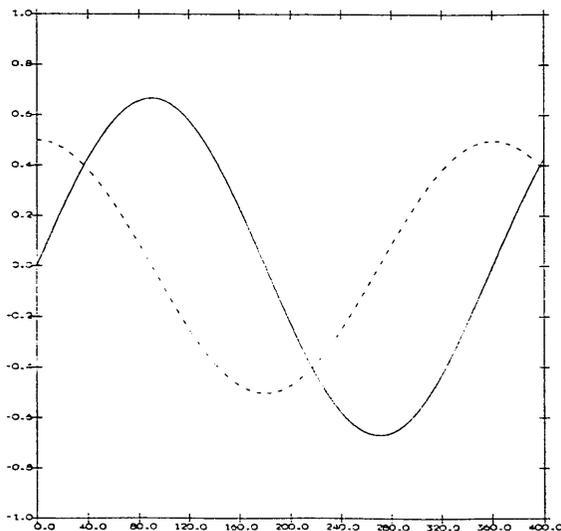


図 3. 3

3. 7 利用者データの再生

メタファイルに格納された利用者データを再生する場合に用いるサブルーチンを表 3. 3 に示す.

表 3. 3 利用者データ再生時に使用するサブルーチン

サブルーチン名	機能
GGTITM	詳細は 3. 5 参照
GRDITM	詳細は 3. 5 参照. 不要なデータを排除する場合に用いる
GUREC	データレコード分解

メタファイルに格納された利用者データは、項目種、項目長などの情報が付加されるが、データそのものは加工されない。ただ、注意しなければならない点は、項目種（利用者データであるから 101 以上の値）、項目長、項目データ（ここでは利用者データ）という利用者データに関する情報だけがメタファイルに入っているわけではない点である。このため、項目を読んで、それが利用者データであるなら読み出し、そうでなければ読み飛ばすという作業をしなくてはならない。読み飛ばす方法は、3. 5 で説明したサブルーチン GRDITM を使用する。このサブルーチンの引数 MIDRL に 0 を指定すれば項目データを読み飛ばし、カレントポイントも更新する。

サブルーチン GRDITM により unnecessary データは読み飛ばし、必要なものだけを読み出すと、読み出された利用者データは、サブルーチン GRDITM の引数 DATREC に格納される。DATREC はデータレコードそのものなので、これを分解する作業をサブルーチン GUREC を用いて行う。以下に呼び出し形式を示す。

```
CALL GUREC(LDR, DATREC, IIL, IRL, ISL, ERRIND, IL, IA, RL, RA, SL, LSTR, STR)
```

引数 LDR は 4 バイトの整数型で、引数 DATREC の次元数を指定する。引数 DATREC はデータの格納されたデータレコードを指定する。この引数は長さ 80 バイトの文字型の 1 次元配列である。引数 IIL は整数データが格納される配列（引数 IA）の次元数を、引数 IRL は実数データが格納される配列（引数 RA）の次元数を、引数 ISL は文字データの格納される配列（引数 STR）と各文字データの長さが格納される配列（引数 LSTR）の次元数を指定する。IIL, IRL, ISL はすべて 4 バイトの整数型で、1 以上を指定する。引数 ERRIND はエラー番号を GKS が通知する領域で、4 バイトの整数型である。IL, RL, SL はそれぞれ、引数 IA, RA, STR に実際に格納したデータの数を GKS が通知する領域で、4 バイトの整数型である。

### 3. 8 利用者データの再生を行うプログラム例

3. 4 でメタファイルに格納した利用者データを再生し、そのデータを使用してグラフを描くプログラムである。このプログラムの出力結果を図 3. 4 に示す。

(プログラム例)

```
0001    INTEGER    ITEM, LENGTH, ERR1, ERR2, RL, IDMMY1, IDMMY2, IDMMY3, IDMMY4
0002    REAL      X(401), Y(401), FX(4), FY(4)
0003    CHARACTER DATREC(21)*80, CDMMY
0004 C
0005    FX(1) =    0.0
0006    FX(2) = 400.0
0007    FX(3) = FX(2)
0008    FX(4) = FX(1)
0009    FY(1) = -1.0
```

解 説

```
0010      FY(2) = FY(1)
0011      FY(3) =  1.0
0012      FY(4) = FY(3)
0013 C
0014      CALL GOPKS (0,1)
0015      CALL GOPWK (1,1,1)
0016      CALL GOPWK (2,2,2)
0017      CALL GACWK (2)
0018 C
0019      1 CONTINUE
0020      CALL GGTITM (1,ITEM,LENGTH)
0021      IF (ITEM.EQ.0) THEN
0022          GOTO 2
0023      ELSEIF (ITEM.EQ.101) THEN
0024          CALL GRDITM (1,LENGTH,21,DATREC)
0025          CALL GUREC (21,DATREC,1,401,1,ERR1,
0026      +              IDMMY1, IDMMY2, RL, X, IDMMY3, IDMMY4, CDMMY)
0027          GOTO 1
0028      ELSEIF (ITEM.EQ.102) THEN
0029          CALL GRDITM (1,LENGTH,21,DATREC)
0030          CALL GUREC (21,DATREC,1,401,1,ERR2,
0031      +              IDMMY1, IDMMY2, RL, Y, IDMMY3, IDMMY4, CDMMY)
0032          GOTO 1
0033      ELSE
0034          CALL GRDITM (1,0,21,DATREC)
0035          GOTO 1
0036      ENDIF
0037 C
0038      2 CONTINUE
0039      CALL GSWN (1,-40.0,440.0,-1.2,1.2)
0040      CALL GSVP (1,0.0,1.0,0.0,1.0)
0041      CALL GSELNT (1)
0042 C
0043      CALL GSLN (1)
0044      CALL GPL (RL,X,Y)
0045      CALL GFA (4,FX,FY)
0046 C
0047      CALL GDAWK (2)
0048      CALL GCLWK (1)
0049      CALL GCLWK (2)
0050      CALL GCLKS
0051 C
0052      STOP
```

0053        END

(プログラムの説明)

- 0001 ~ 0003 必要な変数・配列の宣言。  
 0005 ~ 0012 枠の作画のための座標値の設定。  
 0014 ~ 0017 G K S を開始し、ワークステーション 1 を使用可能状態にし、ワークステーション 2 を出力可能状態にする。  
 0019 ~ 0022 ワークステーション 1 (メタファイルの M I 種別) より項目種と項目長を読む。項目種が 0 ならばデータの終了と見なし、0038 へ分岐する。  
 0023 ~ 0027 101 (メタファイルに格納したとき 101 を指定したため) ならばデータをデータレコードに格納し、そのデータレコードを分解する。  
 0028 ~ 0032 102 (メタファイルに格納したとき 102 を指定したため) ならばデータをデータレコードに格納し、そのデータレコードを分解する。  
 0033 ~ 0036 利用者データ以外 ( $0 \leq \text{項目種} \leq 100$ ) はすべて読み飛ばす。  
 0039 ~ 0041 座標変換。ここでは、正規化変換のみを行い、ワークステーション変換は G K S に任せている。  
 0043 ~ 0045 メタファイルから配列に格納したデータを使用し、曲線を描く。  
 0047 ~ 0050 ワークステーションを使用不可能状態にし、G K S を終了する。

(プログラムの実行)

実行結果をテクトロ端末 (T 4 0 1 4) に表示する。

READY .

GKS85 プログラム名 WS('MI(META2.DATA), 2/2:T4014')

(プログラムの出力結果)

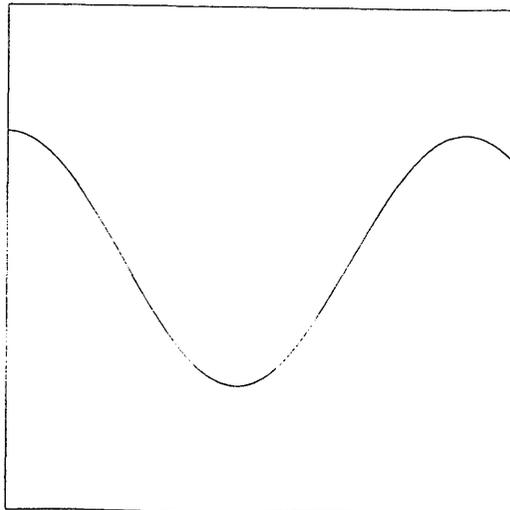


図 3. 4

#### 4. おわりに

GKS入門シリーズは今回が最終回になってしまいますが、GKSには他にもまだいくつかの優れた機能がある。図形入力機能、問合わせ機能等がその代表であるが、これらにほとんど触れずに終わってしまうのは残念である。また、この解説記事に携わったもの全員がGKSを学習しながらの執筆であったため、説明が不足している部分も沢山あったように思う。ただ、プログラム例などは利用者の方が実際に使用されるであろうと思われる形で短いプログラムを示し、そこで説明したサブルーチンのすべてを使ってわかりやすく示したつもりである。

とても使いやすいグラフィックサブルーチンライブラリなので、ぜひ一度利用されることをお勧めする。

#### 参考文献

- [1] GKS入門(1), 九大大型計算機センター広報, Vol.22, No.5, 1989, pp.446 ~ 468
- [2] GKS入門(2), 九大大型計算機センター広報, Vol.22, No.6, 1989, pp.729 ~ 753
- [3] GKS入門(3), 九大大型計算機センター広報, Vol.23, No.1, 1990, pp.001 ~ 013
- [4] GKS入門(4), 九大大型計算機センター広報, Vol.23, No.4, 1990, pp.295 ~ 322
- [5] グラフィックツールによる図形出力ガイド, 九大大型計算機センター広報, Vol.23, No.4, 1990, pp.323 ~ 386
- [6] ルシア マッケイ著, 松本 眞理訳, GKS入門, ノバグラフィックス ジャパン(株).
- [7] 計算機マニュアル, 富士通 GKS85文法書(FORTRAN編)(99SP-8240-1), 富士通(株).
- [8] 計算機マニュアル, 富士通 GKS85使用手引書 V10(99SP-4080-2), 富士通(株).