GUI and Graphics in FORTRAN Programming with Tcl/Tk

Tcl/Tk によるFORTRANプログラミングにおけるGUIとグラフィックス

佐 賀 信 裕 (福岡工業短期大学電子情報システム学科)

GUI and Graphics in FORTRAN Programming with Tcl/Tk

Nobuhiro SAGA (Department of Electronics and information Systems, Fukuoka Junior College of Technology)

Abstract

This paper presents that a graphical user interface (GUI) and a graphics in FORTRAN programming are easily implemented by using Tcl/Tk. The features of these GUI programs are that a look of the GUI, input to FORTRAN program, and a display of output from FORTRAN program are provided by Tcl/Tk, and that the executive routines are provided by FORTRAN.

In graphics, it is shown that for the drawing on the canvas widget in Tcl/Tk, there are two methods; one is the method of using the Xlib in X Window system, the other is that of using the transfer of the x- and y-coordinates of the drawing points; and the plots by the former can be made rapid implementation more than 130 times as compared with that by the latter.

Key words: *FORTRAN, Tcl/Tk, GUI (graphical user interface), graphics, X Window, Xlib*

1. まえがき

最近,プログラミング言語として,C/C++言語やビジュアル系の言語が用いられている。また,FORTRAN (F77) は,以前から数値計算用のプログラミング言語として用いられてきている。しかし,FORTRANでGUI (Graphical User Interface) ベースのプログラムを作成することは困難である。FORTRANでGUIを構築するには,別のライブラリやツール等が必要になる。最近では,GUIベースのアプリケーションが普通になってきている。従って,FORTRANでプログラミング教育を行う場合,学生が容易にGUIベースのプログラムが作成できることが求められる。

本論文では,FORTRANプログラムをビジュアル系言語の一つであるTcl/Tkと連携させることによって,X Windowの環境下で「擬似的」にGUIを備えたプログラム作成が容易にできることを示す。この「擬似的」なGUIプログラムの特徴は,外観やFORTRANプログラムへのデータの入力,並びにFORTRANプログラムからの出力表示等のGUI部分をTcl/Tkで行い,実行処理をFORTRANプログラムが行うことである。こうすることによって,FORTRANプログラムへの入力の変更が容易になり,入力データをいろいろ変えて出力を表示させることが簡単にできる。以上のことは,MS Windowsの環境下でもわずかな修正で動作させることができる。

また,幾つかのまとまった処理を別々のFORTRANプログラムで作成し,コンパイルして実行形式にしておく。Tkから Buttonを押すことによって,一つまたは幾つかの実行プログラムを動作させる。処理全体を一つのプログラムで作成するよりは,それぞれの処理を別々のプログラムで実行させる方が処理ス

平成10年10月31日受付

ピードは速く，また，メモリの使用量も少なくてすむ。実行プログラム間のデータのやりとりは，パイプラインやファイルを介して行うことができる。

さらに，コマンドラインベースの FORTRAN プログラムがほとんど修正なく用いることができるため，以前に作成したプログラムの再利用ができる。

Tcl/TK[1-4]は，John K. Ousterhout によって，1988年カリフォルニア州立大学バークレー校で作られた言語とツールキットである。

また，プログラム等はパシフィック・ハイテック社の Turbo Linux 日本語版2.0で動作確認を行っている。

```
*     巾乗を求める
      PROGRAM  POWER
      REAL   X, BEKIJO
      INTEGER  N, I, M
*
      READ(5,*) X, N
      BEKIJO = 1.0
      IF ( N .NE. 0 ) THEN
         M = IABS(N)
         DO 10 I = 1, M
            BEKIJO = BEKIJO * X
10       CONTINUE
         IF ( N .LT. 0 ) THEN
            BEKIJO = 1.0 / BEKIJO
         END IF
      END IF
      WRITE(6,*) X, ' の', N, ' 乗 = ', BEKIJO
      END
```

図1　べき乗（x^n）を求めるプログラム

```
#!/bin/sh
# the next line restarts using wish \
exec wish "$0" "$@"

       :
       :

set fileName "bekijo"

proc Run { } {
    global  result fileName x n
    set result [exec $fileName << "$x $n\n"]
}

       :
       :

entry .e1 -width 10 -relief sunken -fg red \
        -bg white -textvariable x
entry .e2 -width 5 -relief sunken -fg red \
        -bg white -textvariable n
focus .e1

       :
       :

label .result -width 40 -height 2 -justify left \
        -fg red -bg white -textvariable result \
        -relief groove -borderwidth 4

       :
       :
```

図2　べき乗を求める Tcl/Tk のスクリプト

る。

2．FORTRAN と Tcl/Tk との連携

FORTRAN プログラムと Tcl/Tk との連携の例として，べき乗（xのn乗）を求めるプログラムを考える。FORTRAN プログラムを図1に，Tcl/Tk のスクリプトを図2に示す。また，図2のスクリプトの実行例を図3に示す（10^6を表示）。

スクリプトを実行するためには，図1の FORTRAN プログラムを"bekijo"という実行形式名でコンパイルしておく必要がある。

べき乗プログラムの流れを図4に示す。

2．1　Tk による FORTRAN への入力

Tk（ツールキット）において，キーボードからの文

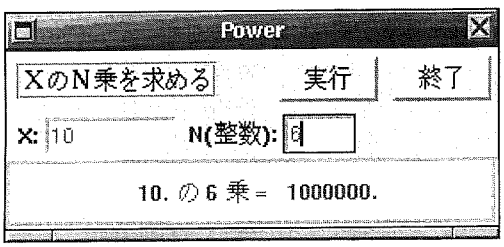

図3　べき乗を求めるTcl/Tkスクリプトの実行結果

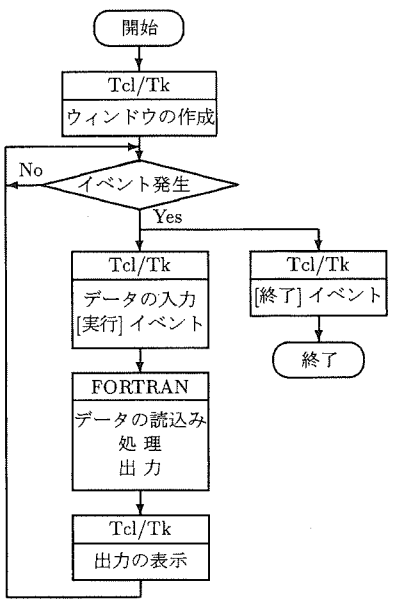

図4　べき乗プログラムの流れ

字の入力は entry widget を用いて行う。入力された値が入る変数を指定するために -textvariable オプションを用いる。変数の値は〝$〟を変数の頭につけることによって取り出すことができる。Tcl スクリプトから UNIX コマンド（実行形式のプログラム）を実行するには式(1)（図2参照）のように exec コマンドを用いる。

```
set result [exec $fileName << "$x $n¥n"]    (1)
```

ここで，fileName は実行形式のプログラム〝bekijo〟を表す。UNIX コマンドへの値の入力は〝<<〟を用いる。二変数以上の場合には，式(1)のように変数名を空白で区切り最後に改行文字〝¥n〟を並べたものを二重引用符で取り囲む。また，〝<〟を用いれば，指定したファイルからデータを UNIX コマンドに入力することもできる。

2．2　FORTRAN から Tk への出力

実行されたプログラム（bekijo）の標準出力は exec コマンドの値として返され，式(1)の場合には変数 result に格納される。この result に格納された値を表示するためには，式(2)（図2参照）のように label コマンドの -textvariable オプションで result を指定する。

```
label  .result -width 40 -height 2 ¥
       -justify left -fg red -bg white ¥
       -textvariable result ¥
       -relief groove -borderwidth 4        (2)
```

また，Tk における表示には listbox や message コマンドを用いることもできる。

3．FORTRAN によるグラフィックス

Tcl/Tk を用いて FORTRAN でグラフィックスをする場合の例として，関数 f（x，z）$=2.0\cdot\cos(\sqrt{x^2+z^2})$ の描画を考える。実行結果を図5に示す。また，その FORTRAN のプログラムを図6に，

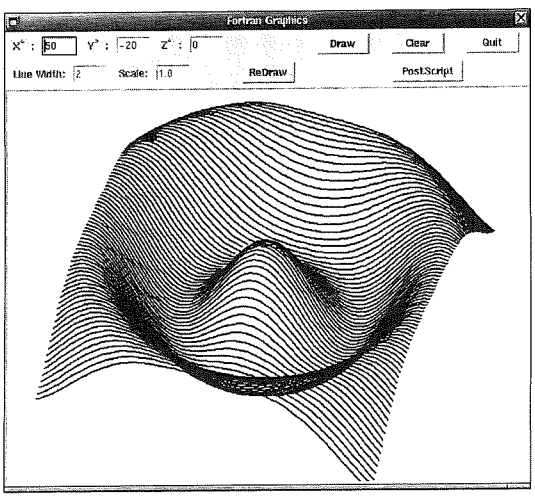

図5　関数 f（x, z）$=2.0\cdot\cos(\sqrt{x^2+z^2})$ を描画する Tcl/Tk スクリプトの実行結果

```
      PROGRAM SMPL
      INTEGER YMAX(0:640), YMIN(0:640)
      INTEGER I, X0, Y0, NX, NY
      INTEGER IX, IZ, S
      REAL F, X, Y, Z, X3, Y3, Z3
      REAL PI, RAD, AX, AY, AZ, XA, YA, ZA
      F(X,Z) = 2.0 * COS(SQRT(X*X + Z*Z))
*
      PI = 4.0 * ATAN(1.0)
      RAD = PI / 180.0
      DO 10 I = 0, 640
         YMAX(I) = -1
         YMIN(I) = 481
 10   CONTINUE
*     原点
      X0 = 320
      Y0 = 240
*     回転角 x軸, y軸, z軸
      READ(5,*) AX, AY, AZ
      XA = AX * RAD
      YA = AY * RAD
      ZA = AZ * RAD
      S = 5
      CALL SETCOL('DARKORANGE')
      DO 20 IZ = 200, -200, -S
         Z = PI * IZ / 100.0
         DO 30 IX = -200, 200, S
            X = PI * IX / 100.0
            Y = F(X, Z)
            CALL ROT(XA,YA,ZA,X,Y,Z,X3,Y3,Z3)
            NX = NINT(X0 + 35 * X3)
            NY = NINT(Y0 - 35 * Y3)
*           陰線処理
            IF (IX .EQ. -200) THEN
               CALL MOVETO(NX, NY)
            ELSE
               IF (NY .LE. YMIN(NX)) THEN
                  YMIN(NX) = NY
                  CALL LINETO(NX, NY)
               ELSE IF (NY .GE. YMAX(NX)) THEN
                  YMAX(NX) = NY
                  CALL LINETO(NX, NY)
               ELSE
                  CALL MOVETO(NX, NY)
               END IF
            END IF
 30      CONTINUE
 20   CONTINUE
      END
```

図6　関数 f（x, z）$=2.0\cdot\cos(\sqrt{x^2+z^2})$ を描画する FORTRAN プログラム

```
#!/bin/sh
# the next line restarts using wish \
exec wish "$0" "$@"

set fileName "smpl"
set WIDTH 640
set HEIGHT 480

canvas .c0 -width $WIDTH -height $HEIGHT \
       -bg white -relief sunken -bd 2

proc DrawGraph { } {
  global result lineColor lineWidth
  foreach  line  [split  $result  \n] {
      scan $line "%d" p
      if { $p == 0 } {
          scan $line "%d %d %d %d %d" p x1 y1 x2 y2
          .c0 create line $x1 $y1 $x2 $y2 \
            -fill $lineColor -width $lineWidth \
            -tag tg0

          :
          :

      } elseif { $p == 9 } {
          scan $line "%d %s" p lineColor
      }
  }
}

button .f0.b0 -text Draw -width 6 -fg black \
 -command {
    set result [exec $fileName << "$xa $ya $za\n"]
    ClearGraph
    DrawGraph
    set scale 1.0
    set scaleOld 1.0
}

          :
          :
```

図7　関数 $f(x,z)=2.0\cdot\cos(\sqrt{x^2+z^2})$ を描画する Tcl/Tk スクリプト

Tcl/Tk のスクリプトを図7に示す。このスクリプトでは視点を変更できるように，Tk から各座標軸の回転角を FORTRAN プログラムに入力するようになっている。

3．1　FORTRAN と Tcl/Tk のインタフェース

この FORTRAN プログラムでは，MOVETO (X, Y)（点(X, Y)に描画点を移動する），LINETO (X, Y)（前の描画点から現在の点(X, Y)に直線を引く），LINE (X_1, Y_1, X_2, Y_2)（点 (Y_1, Y_1) から点 (X_2, Y_2) に直線を引く）サブルーチンが使用できる。また，描画する直線の色も指定できるように SETCOL ('COLOR')（COLOR には色の名前を設定する）サブルーチンも用意する。これらのサブルーチンをまとめて1つのサブルーチン DRAWGR を作成する(図8)。これらサブルーチンからの出力を番号で区別する。ここでは，LINETO と LINE は0，SETCOL は9としている。この番号に従って FORTRAN からの出力が図7の Tcl/Tk のスクリプトに読み込まれる。他の描画サブルーチンが必要な場合は，DRAWGR に追加すればよい。また，同じ直線を2度描画しないように同じ座標点対は出力させないようにしている。

```
      SUBROUTINE DRAWGR
      INTEGER  X, Y, XX, YY, X1, Y1, X2, Y2
      INTEGER  XX1, YY1, XX2, YY2
      CHARACTER COLOR*(*), CC*20
      SAVE  XX, YY, CC, XX1, YY1, XX2, YY2
      DATA  CC/ 'COLOR'/
      DATA  XX, YY/ -9999, -9999 /
      DATA  XX1, YY1, XX2, YY2/ 4*-9999 /
            :
            :
      ENTRY  MOVETO(X, Y)
      IF ((X .EQ. XX) .AND. (Y .EQ. YY)) THEN
         RETURN
      ELSE
         XX = X
         YY = Y
      END IF
      RETURN
*
      ENTRY  LINETO(X, Y)
      IF ((X .EQ. XX) .AND. (Y .EQ. YY)) THEN
         RETURN
      ELSE
         WRITE(6,*) 0, XX, YY, X, Y
         XX = X
         YY = Y
      END IF
      RETURN
*
      ENTRY LINE(X1, Y1, X2, Y2)
      IF ((X1.EQ.XX1) .AND.(Y1.EQ.YY1) .AND.
     +    (X2.EQ.XX2) .AND.(Y2.EQ.YY2)) THEN
         RETURN
      ELSE
         WRITE(6,*) 0, X1, Y1, X2, Y2
      END IF
      XX1 = X1
      YY1 = Y1
      XX2 = X2
      YY2 = Y2
      RETURN
*
      ENTRY  SETCOL(COLOR)
      IF (COLOR .NE. CC) THEN
         WRITE(6,*) 9, COLOR
         CC = COLOR
      END IF
      RETURN
      END
```

図8　FORTRAN と Tcl/Tk とのインタフェースのサブルーチン DRAWGR

3．2　Tk での描画

Tcl/Tk で描画する場合，Tk の canvas コマンドを用いる。この canvas コマンドで作成されたキャンバスに canvas items（arc, bitmap, image, line, oval, polygon, rectangle, text, window）の中の line (直線)アイテムを用いて描画する。直線を引く line を

```
void gopen_()
{
   long winID;

/* ディスプレイを開く                    */
   fgl_d = XOpenDisplay(NULL);
/* ルートウィンドウの ID を取得する      */
   fgl_rw = XDefaultRootWindow(fgl_d);
/* ウィンドウの ID を Tk から取得する */
   scanf("%x", &winID);                          (7)
/* ウィンドウ型の ID に変換する          */
   fgl_w = (Window)winID;                        (8)
           :
           :
/* ウィンドウをディスプレイに表示する */
   XMapWindow(fgl_d, fgl_w);                     (9)
           :
           :
/*
   リクエスト・バッファの中身を強制送出する
*/
   XFlush(fgl_d);
}
```

図9　FORTRAN でグラフィックスを使用するための Xlib の初期設定関数

用いる場合，始点と終点の座標が必要になる。この始点と終点の座標は3．1節で示したように FORTRAN で計算し，その結果を Tk に取り込み（式(3)），式(4)の line アイテムを用いて，.c0 create コマンドでキャンバスに描画する。

```
set result [exec $fileName << "$xa $ya $za¥n"]
foreach line  [split $result ¥n]  {
  scan $line "%d %d %d %d" x1 y1 x2 y2          (3)
  .c0 create line $x1 $y1 $x2 $y2 ¥
    -fill $lineColor -width $lineWidth ¥
    -tag tg0                                    (4)
```

4．Xlib と Tcl/Tk によるグラフィックス

第3章では，FORTRAN プログラムで描画点を出力し，その出力データを Tk の canvas widget に取り込み，line アイテムでキャンバスにグラフを作図した。しかし，この方法では，描画速度が遅くなる。そこで，Xウィンドウ・バージョン11（X11と書く）のライブラリ Xlib[5]を用いて直接キャンパスに描画する方法を考える。すなわち，この方法は3章で作成した FORTRAN プログラムをそのまま用い，FORTRAN と Tk のインタフェースとして X11の Xlib を用いることである。そのためには，Xサーバから取得した Tk の canvas widget の ID（固有番号）が必要になる。これは式(5)に示すように winfo コマンドを用いて取得することができる。

```
set winID  [winfo id .c0]                       (5)
```

このウィンドウ ID (winID) を次式のように示すように fileName で表された実行プログラムに入力する。

```
exec $fileName  << "$winID"                     (6)
```

この winID を図9に示す FORTRAN でグラフィックスを使用するための初期設定関数 gonen_（　）中の式(7)で取得する。更に，式(8)でウィンドウ型の ID に変換し，ウィンドウをディスプレイにマップする（表示する）XMapWindow 関数の引数として入力する（式(9)）。

このように X11のライブラリ Xlib を用いることによって，3章と同じグラフの描画で Tcl/Tk コマンドの〝clock clicks〟を用いて計測した結果，3章の描画方法より130倍以上の速度の改善が得られた。

5．む　す　び

第2章では，FORTRAN と Tcl/Tk とを連携させることによって，FORTRAN の不得意な GUI 部分を Tcl/Tk で作成し，得意な計算の実行処理は FORTRAN で行うことによって，学生が容易に GUI ベースのプログラムが作成できることを示した。また，第3章では，FORTRAN と Tcl/Tk とのインタフェース（描画点のｘ，ｙ座標を出力する）を作成することによって，FORTRAN で GUI ベースのグラフィクスができることを示した。この方法は，わずかな修正だけで MS Windows 環境下でも有効である。更に，第4章では，FORTRAN と Tcl/Tk とのインタフェースとして X11のライブラリ Xlib を用いることによって，第3章で示したグラフの描画速度を130倍以上に速めることができることを示した。

以上，FORTRAN と Tcl/Tk を連携することによって，「擬似的」ではあるけれども容易に GUI を備えたプログラムが作成できることを示した。

今後の課題として，FORTRAN と Tcl/Tk を連携させるためのインタフェース機能の強化と，ライブラ

リ化することが必要と考える。

参 考 文 献

(1) J. K. Ousterhout: Tcl and the Tk Toolkit (Addison-Wesley, 1994).
(西中芳幸，石曽根　信訳：Tcl&Tk ツールキット，ソフトバンク，平成7年.)

(2) 須栗歩人：入門 Tcl/Tk，秀和システム，平成10年.

(3) B. B.ウェルチ：Tcl/Tk 入門，プレンティスホール出版，平成9年.

(4) 久野　靖：入門 tcl/tk，アスキー出版局，平成9年.

(5) 木下凌一，林　秀幸：X-Window Ver. 11 プログラミング，第2版，日刊工業新聞社，平成5年.