ISOSRF, A PACKAGE OF ROUTINES USED TO DRAW ISOSURFACES

David J. Kennison
NCAR, P.O. Box 3000, Boulder, Colorado 80307-3000
email: kennison@ncar.ucar.edu (128.117.64.4)

ISOSRF, A PACKAGE OF ROUTINES USED TO DRAW ISOSURFACES

David J. Kennison
NCAR, P.O. Box 3000, Boulder, Colorado 80307-3000
email: kennison@ncar.ucar.edu (128.117.64.4)

1.0 INTRODUCTION

The graphics package ISOSRF allows the user to draw a perspective view of the surface defined by the equation $f(u,v,w)=f_{iso}$, where the function f is approximated by a 3-dimensional array of data and f_{iso} is a user-specified “isovalue”. Contours created by taking slices in any of three directions through the surface are drawn with hidden portions of the contour lines removed.

Either of the routines EZISOS or ISOSRF may be called to draw an isosurface. The former has a simpler argument list, but makes certain assumptions about what is to be done.

Routines with names of the form ISGETx, where “x” stands for “I” or “R”, are used to retrieve the current values of various internal parameters affecting the behavior of ISOSRF.

Routines with names of the form ISSETx, where “x” stands for “I” or “R”, are used to give new values to internal parameters.

Routines mentioned above are discussed in detail in the section named “ROUTINES”. Internal parameters are discussed in the section named “INTERNAL PARAMETERS”.

2.0 ROUTINES

2.1 EZISOS (F,MU,MV,MW,EYE,SLAB,FISO)

Draws an isosurface.

2.1.1 Usage

Use the FORTRAN statement

```
CALL EZISOS (F,MU,MV,MW,EYE,SLAB,FISO)
```

to draw an isosurface if

- all of the input array F is to be used,
- ISOSRF's argument IFLAG is to be chosen internally, and
- a frame advance is to be done after the isosurface is drawn.

If one or more of these conditions is not met, use the routine ISOSRF (which see, below) instead.

2.1.2 Arguments

F (an input array of type REAL, dimensioned MU x MV x MW) is a three-dimensional array of data defining the function f(u,v,w). The entire array (elements F(IU,IV,IW), for IU = 1 to MU, IV = 1 to MV, and IW = 1 to MW) is to be used. The data are considered to lie in a box in 3-space with opposite corners (in the UVW coordinate system) (1.,1.,1.) and (REAL(MU),REAL(MV),REAL(MW)). The element F(IU,IV,IW) is considered to be the value of the function f at the point (REAL(IU),REAL(IV),REAL(IW)).

MU (an input expression of type INTEGER) is the first dimension of the array F.

MV (an input expression of type INTEGER) is the second dimension of the array F.

MW (an input expression of type INTEGER) is the third dimension of the array F.

EYE (an input array of type REAL, dimensioned 3) is the position of the eye in the UVW coordinate system. The eye position, at (EYE(1),EYE(2),EYE(3)), must be outside the box containing the data. The point being looked at is at the center of the data box and the projection plane is perpendicular to the line of sight. While gaining experience with the routine, a good choice for EYE is (5.*REAL(MU),3.5*REAL(MV),2.*REAL(MW)).

SLAB (a scratch array of type REAL, dimensioned at least n x n, where "n" is defined to be MAX[MU,MV,MW]+2) is a workspace for ISOSRF.

FISO (an input expression of type REAL) is the value of f_{iso} in the equation $f(u,v,w)=f_{iso}$, which defines the isosurface to be drawn.

2.2 ISOSRF (F,LU,MU,LV,MV,MW,EYE,MUVWP2, . . .)

(The remaining arguments are **SLAB**, **FISO**, and **IFLAG**.)

Draws an isosurface.

2.2.1 Usage

Use the FORTRAN statement

```
CALL ISOSRF (F,LU,MU,LV,MV,MW,EYE,MUVWP2,SLAB,FISO,IFLAG)
```

to draw an isosurface in cases in which EZISOS (which see, above) cannot be used.

2.2.2 Arguments

F (an input array of type REAL, dimensioned LU x LV x m, where “m” is greater than or equal to MW) is a three-dimensional array of data defining the function f(u,v,w). Only the portion of the array consisting of elements F(IU,IV,IW), for IU = 1 to MU, IV = 1 to MV, and IW = 1 to MW, is to be used. This may or may not be the entire array. The data are considered to lie in a box in 3-space with opposite corners (in the UVW coordinate system) (1.,1.,1.) and (REAL(MU),REAL(MV),REAL(MW)). The element F(IU,IV,IW) is considered to be the value of the function f at the point (REAL(IU),REAL(IV),REAL(IW)).

LU (an input expression of type INTEGER) is the first dimension of the array F.

MU (an input expression of type INTEGER) defines the range to be used for the first subscript of the array F.

LV (an input expression of type INTEGER) is the second dimension of the array F.

MV (an input expression of type INTEGER) defines the range to be used for the second subscript of the array F.

MW (an input expression of type INTEGER) defines the range to be used for the third subscript of the array F.

EYE (an input array of type REAL, dimensioned 3) is the position of the eye in the UVW coordinate system. The eye position, at (EYE(1),EYE(2),EYE(3)), must be outside the box containing the data. The point being looked at is at the center of the data box and the projection plane is perpendicular to the line of sight. While gaining experience with the routine, a good choice for EYE is (5.*REAL(MU),3.5*REAL(MV),2.*REAL(MW)).

MUVWP2 (an input expression of type INTEGER) has the value MAX(MU,MV,MW)+2.

SLAB (a scratch array of type REAL, dimensioned at least MUVWP2 x MUVWP2) is a workspace for ISOSRF.

FISO (an input expression of type REAL) is the value of f_{iso} in the equation $f(u,v,w)=f_{iso}$, which defines the isosurface to be drawn.

IFLAG (an input expression of type INTEGER) serves two purposes:

- The absolute value of IFLAG determines which type of lines are drawn to approximate the isosurface. Three types of lines are considered: lines of constant U, lines of constant V, and lines of constant W. The following table lists the types of lines drawn for various values of ABS(IFLAG):

ABS(IFLAG)	Lines of Constant U	Lines of Constant V	Lines of Constant W
1	no	no	yes
2	no	yes	no

ABS(IFLAG)	Lines of Constant U	Lines of Constant V	Lines of Constant W
3	no	yes	yes
4	yes	no	no
5	yes	no	yes
6	yes	yes	no
7	yes	yes	yes

- The sign of IFLAG determines what is inside and what is outside the solid bounded by the isosurface and thus which lines are visible and what is done at the boundary of the box containing the data. If IFLAG is positive, values greater than FISO are considered to be inside the solid formed by the isosurface. If IFLAG is negative, values less than FISO are considered to be inside the solid formed by the isosurface. If the algorithm draws a cube, reverse the sign of IFLAG.

2.3 ISGETI (CNP,IVP) and ISGETR (CNP,RVP)

These routines are used to retrieve the current values of internal parameters.

2.3.1 Usage

Use the FORTRAN statement

```
CALL ISGETx (CNP,xVP)
```

where “x” is an “I” or an “R” (depending on the type of the value involved), to retrieve, in the variable xVP, the value of the parameter specified by CNP.

For descriptions of all the internal parameters, see the section named “INTERNAL PARAMETERS”.

2.3.2 Arguments

CNP (an input constant or variable of type CHARACTER) is the name of an internal parameter. Only the first two characters of the name are meaningful, but a longer character string may be used to more clearly document what internal parameter is being retrieved.

xVP (an output variable of type INTEGER or REAL, depending on the value of “x”) is the name of a variable into which the value of the parameter specified by CNP is to be retrieved.

2.4 ISSETI (CNP,IVP) and ISSETR (CNP,RVP)

These routines are used to reset the current values of internal parameters.

2.4.1 Usage

Use the FORTRAN statement

```
CALL ISSETx (CNP,xVP)
```

where “x” is an “I” or an “R” (depending on the type of the value involved), to set the value of the parameter specified by CNP to the value of the variable xVP.

For descriptions of all the internal parameters, see the section named “INTERNAL PARAMETERS”.

2.4.2 Arguments

CNP (an input constant or variable of type CHARACTER) is the name of an internal parameter. Only the first two characters of the name are meaningful, but a longer character string may be used to more clearly document what internal parameter is being set.

xVP (an input expression of type INTEGER or REAL, depending on the value of “x”) is the desired new value of the parameter specified by CNP. This value will be used until the next call resetting it.

3.0 INTERNAL PARAMETERS

The internal parameters of ISOSRF are described in the following table.:

Name	Type	Description
'IU'	I	Number of extra slabs to be interpolated between each pair of slabs perpendicular to the U axis. Using a non-zero value takes longer, but makes a better picture. Negate the value to interpolate the extra slabs but not draw them, which has the effect of reducing errors due to being able to see between slabs. Attempting to give 'IU' a value less than -10 will give it the value -10; similarly, trying to give it a value greater than 10 will give it the value 10. The default value is 0.
'IV'	I	Like 'IU', but applies to extra slabs perpendicular to the V axis. The default value is 0.
'IW'	I	Like 'IU', but applies to extra slabs perpendicular to the W axis. The default value is 0.
'RF'	I	Controls drawing of reference planes and axes. Zero means “don’t draw them”; non-zero means “draw them”. The default value is 1.
'RS'	R	Zero turns off the “relative size” feature; the 3D grid box will be scaled to fill almost the entire plotter frame. Give 'RS' a non-zero value to turn on the “relative size” feature and to specify that distance from which the box, when viewed from the direction which makes its image largest, almost fills the frame. Views from closer in will give a bigger image and views from farther out will give a smaller image. This is useful when making movies with the “camera” flying around the box. The default value is 0.

Name	Type	Description
'SL'	R	Segment length. When contours are smoothed, this parameter specifies the approximate length of the line segments used to draw the smooth curves. Values outside the range from .0001 to .1 will be mapped to the nearer end of that range. The default value is .01.
'SM'	I	Screen model selector. Zero selects the coarse screen model (128 x 128); non-zero selects a finer screen model (256 x 256). Using the former is faster, but using the latter gives better pictures. The default value is 0.
'ST'	R	Spline tension. If zero, turns off the smoothing of contours. If non-zero, turns the smoother on and determines the tension on the splines used to do the smoothing. Use of values greater than about 15. has been observed to cause overflow on some machines. The default value is 0.
'SV'	I	Special value. The value zero turns off the special-value feature. A non-zero value turns on the feature and specifies the special value itself. Regions filled with this value are treated as being outside the volume bounded by the isosurface. Preliminary experiments seem to indicate that sometimes the use of special values works well and sometimes it doesn't (depending on the pattern of the special values in the field). The default value is 0.
'VB'	R	The position of the bottom edge of the viewport in which the isosurface is to be displayed, expressed as a fraction between 0 (the bottom edge of the frame) and 1 (the top edge of the frame). The default value is 0.
'VL'	R	The position of the left edge of the viewport in which the iso-surface is to be displayed, expressed as a fraction between 0 (the left edge of the frame) and 1 (the right edge of the frame). The default value is 0.
'VR'	R	The position of the right edge of the viewport in which the iso-surface is to be displayed, expressed as a fraction between 0 (the left edge of the frame) and 1 (the right edge of the frame). The default value is 1.
'VT'	R	The position of the top edge of the viewport in which the iso-surface is to be displayed, expressed as a fraction between 0 (the bottom of the frame) and 1 (the top edge of the frame). The default value is 1.

4.0 ERROR MESSAGES

The error messages described below can be written to the output unit by routines in the package ISOSRF. All of these errors are fatal and cause the user's program to terminate execution (on the other hand, all of the errors are easily avoided).

```
ISGETI OR ISGETR - PARAMETER NAME TOO SHORT - x
```

The argument "PNAM", in a call to ISGETI or ISGETR, is only one character long.

```
ISGETI OR ISGETR - PARAMETER NAME NOT KNOWN - xx
```

The argument “PNAM”, in a call to ISGETI or ISGETR, begins with two characters that do not constitute one of the legal internal parameter names described in the section “INTERNAL PARAMETERS”.

```
ISSETI OR ISSETR - PARAMETER NAME TOO SHORT - x
```

The argument “PNAM”, in a call to ISSETI or ISSETR, is only one character long.

```
ISSETI OR ISSETR - PARAMETER NAME NOT KNOWN - xx
```

The argument “PNAM”, in a call to ISSETI or ISSETR, begins with two characters that do not constitute one of the legal internal parameter names described in the section “INTERNAL PARAMETERS”.

```
ISINIT - INSTALLATION OF ISOSRF IMPROPERLY DONE - SEE DIRECTIONS IN CODE
```

This error message means that the screen models used to do hidden-line removal have been found to be improperly dimensioned for the system on which the code is being run. About the only way this can happen is if the distributed code is run on a a machine with sixteen-bit words or if the distributed code has been modified incorrectly.

5.0 EXAMPLES

On a Unix system on which NCAR Graphics has been installed, the command “ncargex” may be used to acquire the code for and run an example for ISOSRF; the name of this example is “tisosr”; the command required to get and run the example is therefore

```
ncargex tisosr
```

To acquire the code, but not run it, use the command

```
ncargex -n tisosr
```

Then, after modifying the example code as desired, compile and run it with the commands

```
ncargf77 *.f
a.out
```

View the metafile with the command

```
ctrans gmeta
```

More information may be found in the document “Using NCAR Graphics in a Unix Environment”.

6.0 OTHER INFORMATION

6.1 Entry Points

Principal routines: ISOSRF and EZISOS.

Parameter-access routines: ISGETI, ISGETR, ISSETI, and ISSETR.

Routines which are not currently advertised as user-callable, but may be of interest to a knowledgeable user: ISPLTF and ISTR32.

Strictly internal routines: ISFILL, ISGFXY, ISINIT, ISSMTH, ISST3D, ISTRCL, and ISZERO.

A block data routine, in which default values of internal parameters are set: ISBLDA.

Also supported for historical reasons: DRAWI, FRSTC, and TRN32I.

6.2 Common Blocks

Common block names used in ISOSRF are ISCOMN, PWRZ1I, and TEMPRX.

6.3 Required Library Routines

ISOSRF uses the ERPRT77 package and the SPPS.

6.4 History

ISOSRF was developed for users of ISOSRFHR with smaller arrays. In April and May, 1991, some bugs were fixed and some improvements were made.

6.5 Algorithm

Cuts through the three-dimensional array are supplied with boundary values which ensure that all contour lines will be closed and then contoured by a routine which also marks a bit-map model of the plotter frame. (Smoothing is optional.) The interiors of the contours in the model are then filled in and the result is ORed into a second model of the plotter frame that is used to determine the visibility of subsequent contour lines.

The hidden-line algorithm is not exact, so visibility errors can occur.

6.6 Miscellany

Space requirements for ISOSRF can be reduced by changing the sizes of the arrays ISCR and ISCA (found in the common block ISCOMN). These need to hold 512x256 bits apiece, so, on a 64-bit machine, ISCR and ISCA can be dimensioned 8x256 instead of the default 16x256. On a 32-bit machine, 16x256 will suffice. On a 16-bit machine, the dimensions must be increased to 32x256. If the first dimensions of ISCR and ISCA are changed, the value of the common variable LX (set in ISBLDA) must be changed to match.

Transformations can be achieved by adjusting scaling statement functions in ISOSRF, ISST3D, and ISGFXY.

Three-dimensional perspective character labeling of ISOSRF is possible by using the utility PWRZI. For a description of its usage, see the PWRZI documentation.