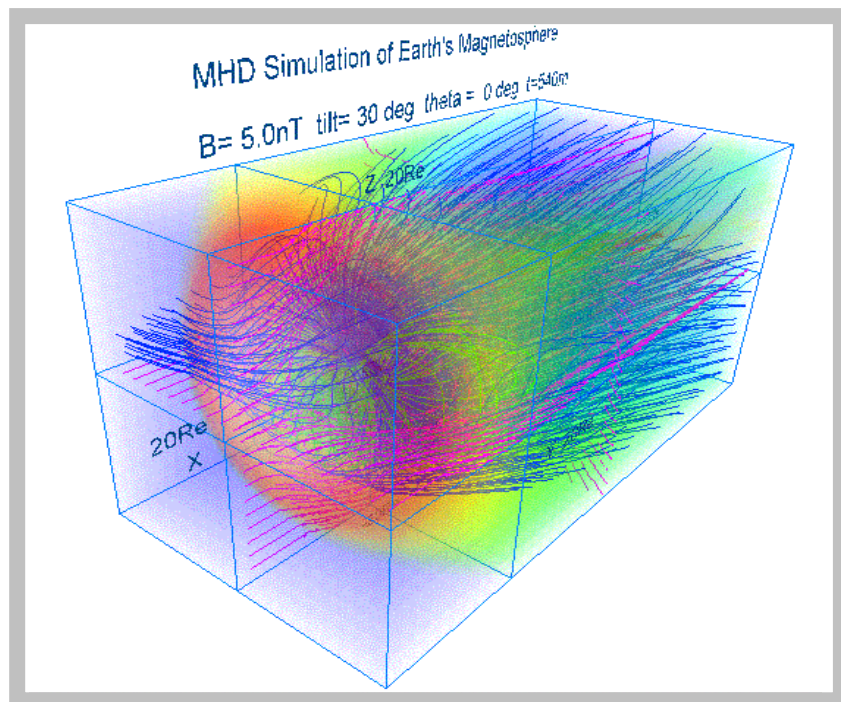


JST Computer Science and Technology Promotion Program
"The Development of Space Simulation Net-Laboratory System"

Grants-in-Aid from the Japan Society for Promotion Science (JSPS) (A)(1)
"Study of Solar Wind-Magnetosphere-Ionosphere Dynamics Using
Common Parallel Computation Codes in MHD and Particle Simulation"

How to Use VRML (Using the Fortran and C)



March, 2004

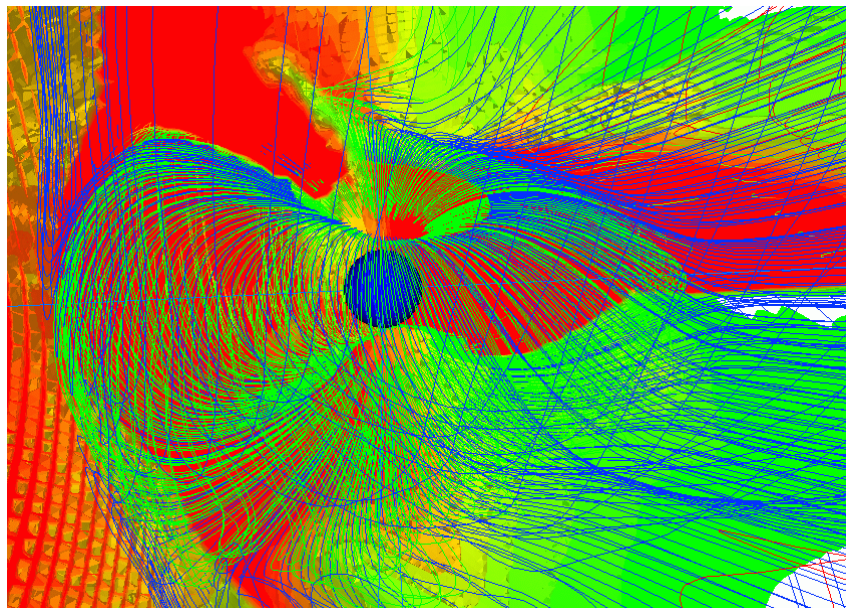
Solar-Terrestrial Environment Laboratory, Nagoya University

Tatsuki Ogino

JST Computer Science and Technology Promotion Program
"The Development of Space Simulation Net-Laboratory System"

Grants-in-Aid from the Japan Society for Promotion Science (JSPS) (A)(1)
"Study of Solar Wind-Magnetosphere-Ionosphere Dynamics Using
Common Parallel Computation Codes in MHD and Particle Simulation"

How to Use VRML (Using the Fortran and C)



March, 2004

Solar-Terrestrial Environment Laboratory, Nagoya University

Tatsuki Ogino

Index

Purpose	1
Content	1
1. Fundamental example (1)	4
1-1. Cube	5
1-2. Rectangular parallelepiped	6
1-3. Trigonal pyramid	7
1-4. Column	8
1-5. Sphere	9
1-6. Background color	10
1-7. Character string	11
1-8. Polygonal line : monochrome	12
1-9. Polygonal line : gradation	13
1-10. Dotted line : monochrome	14
1-11. Dotted line : gradation	15
1-12. Arrow with the thickness	16
1-13. Curve with the thickness	17
1-14. Plane (1) : Monochromatic triangular mesh	18
1-15. Plane (2) : Triangular belt	19
1-16. Plane (3) : Polychrome triangular mesh	20
1-17. Solid : It is composed of the triangular mesh	21
1-18. Texture mapping	22
1-19. The complicated curved surface : Triangular mesh	23
2. Fundamental example (2)	24
2-1. Character	25
2-2. Point	26
2-3. Line	27
2-4. Triangular mesh	28
2-5. Pixel image	29
2-6. Frame and pixel image and character	30

2-7. Equivalence plane	31
2-8. Outer magnetosphere by three-dimensional lattice	32
2-9. Inner magnetosphere by three-dimensional lattice	33
2-10. Ionosphere by three-dimensional lattice	34
2-11. Unification of outer / inner magnetosphere and inosphere	35
3. Application to the MHD simulation	36
3-1. Sectional views such as plasma temperature of the earth's magnetosphere are drew by the pixel image	37
<The pixel image using maximum value and minimum value>	37
3-2. Three-dimensional structure of the earth's magnetosphere	39
<The decision by beforehand describing line of magnetic force>	39
<Three-dimensional line of magnetic force starting from polar region>	40
<Three-dimensional line of magnetic force starting from equatorial plane>	41
<The test of the boundary in the closed region with that it opens>	42
3-3. The synthesis of the three-dimensional image of pixel image and line of magnetic force	43
3-4. Three dimensional distribution of the plasma temperature in the whole earth's magnetosphere drawn by the multiple pixel image planes	44
4. 3-Dimensional MHD Simulation of Earth's Magnetosphere	46
5. Visualization to Various 3-Dimensional MHD Models of Earth's Magnetosphere	51
5-a. Half volume model of earth's magnetosphere with IMF By and Bz components	52
5-b. Quarter volume model of earth's magnetosphere	54
5-c. Half volume model of earth's magnetosphere with dipole tilt	57
5-d. Whole volume model of earth's magnetosphere	59
6 . Application of VRML to 3-Dimensional MHD Models of Earth's Magnetosphere	61
6-a. 3-dimensional visualization of earth's magnetosphere with multiple planes	61
6-b. 3-dimensional visualization of earth's magnetosphere with equivalence planes	63
6-c. Cropping of 3-dimensional data	65
7 . 3-dimensional visualization of scalar physical quantity with isosurfaces by C language	66
References	69
Fortran program and image file list	70

Purpose

Three-dimensional visualization is indispensable in order to understand three-dimensional simulation results such as the earth's magnetosphere better. By appearance of the VRML (Virtual Reality Modeling Language) which is only the international standard language for the Internet's three-dimensional visualization, it would be possible to watch the three-dimensional image as VRML file by the VRML viewer, even if people does not have three-dimensional image processing special computers and three-dimensional image processing special software. However, utilization and popularization of the VRML do not advance very much in present state, because that there is no a convenient tool which easily makes the VRML file and that it is not easy to make the complicated VRML file.

In order to develop the method for visualizing three-dimensional simulation result of the magnetosphere by using the VRML, we have wrestled in making the VRML file earlier by utilizing experience and knowledge cultivated for a long time for the image processing using Fortran and C language. In the computer science and technology project as the extension, we have made this manual "How to Use VRML" for the purpose of the three-dimensional visualization of the simulation results by presenting interface subroutine packages for the VRML contents preparation by Fortran and C language, and we offer the privilege to the simulation and data analysis scientists by advancing development and standardization of the visualization tool.

Content

Three-dimensional visualization is necessary in order to know what has obtained in the three-dimensional and global MHD simulation of the solar wind and earth's magnetosphere interaction. For example, movement of the image and rotation of coordinate axis are methods for better understandings of three-dimensional magnetic field lines. Though three-dimensional visualization tools such as AVS of the marketing, etc. are usually convenient and worthwhile, they often leave the three-dimensional image display the part which is not sufficient at any cost, when it intends to describe the figure really described, because it is too rich in a diversity. In such case, basic program of the image processing must be composed. We have carried out this development by using three-dimensional graphics machinery TITAN and Indigo-2 and three-dimensional graphics software Dore, AVS, Open-GL. Using three-dimensional image processing function of Z-buffers of the special-purpose machinery,

etc., by rotating the object immediately, and doing the scaling magnetic field lines must be described in the three dimensional space, and by choosing and seeable viewpoint, it has been utilized for the understanding of the three-dimensional structure.

However, the situation has completely changed by appearance of the VRML, that is, it is not necessary for the three-dimensional image processing to have three-dimensional graphics machinery and three-dimensional image processing software. The speed of the three-dimensional image processing (rotation, scaling, etc.) is decided with the dependence on the ability of the computer, and viewers such as Cosmo Player of the VRML2.0 correspondence have standardly been added, if browsers such as recent Netscape and Internet Explorer are used. The personal computer became also recently high-speed, and the performance which is not inferior to Indigo-2 made by SGI, etc. can be demonstrated, if high-speed cpu and graphic accelerator are piled, and if the more and more sufficient memory (over 128 MB) is put. And, we can comfortably observe the three-dimensional image with the high accuracy by using Webspase and Cosmo Worlds of SGI.

How the preparation of the VRML file is realized, is described in the following. We prepare the Fortran Interface Subroutine Package for the VRML file preparation, and VRML file (*.wrl) is directly made from three-dimensional simulation data using the Fortran program. Though there is a difference between two-dimension and three-dimension, this method is similar to the method for making the PostScript image file. In the viewer of the VRML, there generally "walk mode" and "examine mode". The walk mode moves the viewpoint, and the examine mode does transfer and rotation and scaling of the object. It is possible to examine more apparently the three-dimensional structure of the magnetosphere in this way. Present important problem is for the VRML files to grow file size up to several MB, since VRML file of the three-dimensional magnetic field lines drawing of the earth's magnetosphere uses the ascii file. By using the VRML binary file or compression of the ascii VRML file, it is also expected that it is considerably improved this problem. It is future most interesting problem to display the three-dimensional moving image of temporal evolution of the earth's magnetosphere with the change of IMF in the VRML, namely a real three-dimensional animation movie.

This "How to Use VRML by Fortran: Method and Contents" includes the following content. In chapter 1, chapter 2, the fundamental examples which makes the three-dimensional image file of the VRML2.0 type by using the Fortran program, and examples which can be utilized as

it is for the analysis of simulation result are shown. In addition, the specific example in applying to the MHD simulation result of solar wind and earth's magnetosphere interaction is shown in chapter 3. In chapter 4 is shown a series of specific example which demonstrated how image analysis in the three-dimensional global MHD simulation and the image processing by Postscript and VRML in the earth's magnetosphere are executed. In final chapters 5 and 6 are shown practical VRML examples for visualization of the earth's magnetosphere simulated by various 3-dimensional MHD models. And, CD-ROM is putting all main Fortran programs and subroutine packages shown in this manual as a source program format. We are happy if it is useful for the simulation and data analysis scientists.

1. Fundamental example (1)

In this chapter, the fundamental example which makes the image file of the VRML2.0 type using the Fortran program is shown. Concretely, the image file of the VRML2.0 type is made by compiling, linking and executing each main program and subroutine package.

Usage:

1. `f77 -c -O pk1subrtn.f`

The object module (pk1subrtn.o) is made by the compilation of subroutine package (pk1subrtn.f).

2. `f77 -O box1.f pk1subrtn.o`

The execution file (a.out) is made by the link of the main program (box1.f etc.) with the object module (pk1subrtn.o).

3. `a.out`

The VRML 2.0 (fort.10) file is made by the execution of the execution file (a.out).

4. `mv fort.10 box1.wrl`

The file name of made VRML is changed at "***.wrl".

1-1. Cube

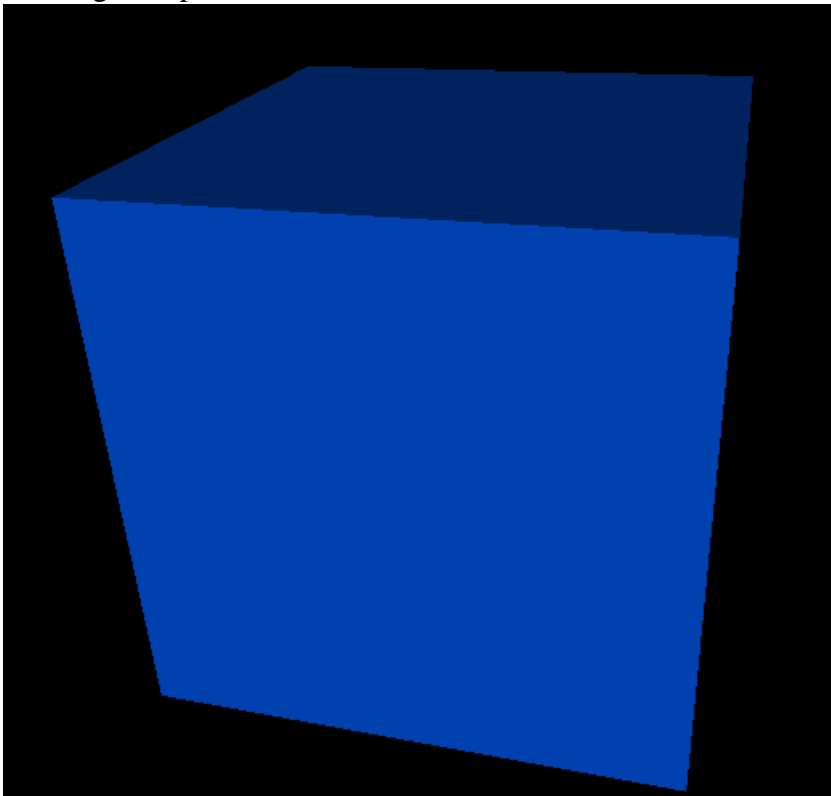
main program box1.f
subroutine box1 (r, g, b, sz)

Description: This program draws a cube by giving the color value and the length of one side.

Input variable:

r : The red value [0,1]
g : The green value [0,1]
b : The blue value [0,1]
sz : The length of one side of the cube

Drawing example: box1.wrl



1-2. Rectangular parallelepiped

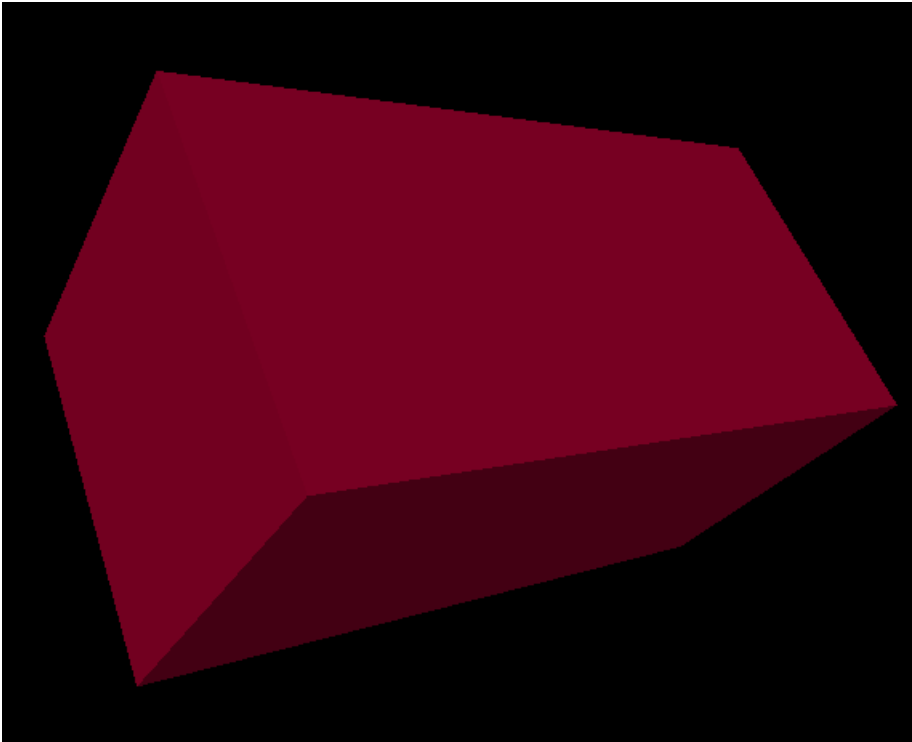
main program box2.f
subroutine box2 (r, g, b, sx, sy, sz)

Description: This program draws a rectangular parallelepiped by giving the color value and the length of three edges.

Input variable:

r : The red value [0,1]
g : The green value [0,1]
b : The blue value [0,1]
sx : The length of one side of the x direction
sy : The length of one side of the y direction
sz : The length of one side of the z direction

Drawing example: box2.wrl



1-3. Trigonal pyramid

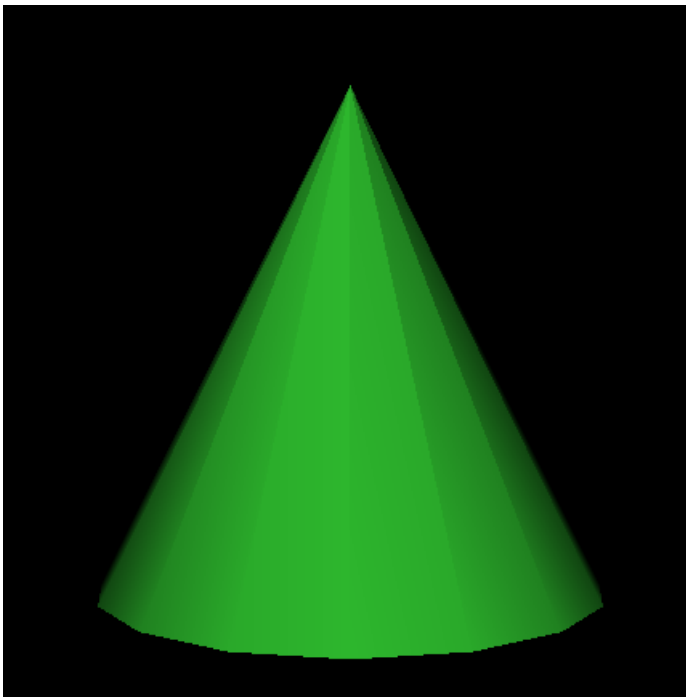
main program cone1.f
subroutine cone1k1 (rd, h, r, g, b, x, y, z)

Description: This program draws the trigonal pyramid by giving the color value and the radius of the basal plane, the height.

Input variable:

rd : The radius of the basal plane
h : The height
r : The red value [0,1]
g : The green value [0,1]
b : The blue value [0,1]
x : Parallel translation quantity of the x direction
y : Parallel translation quantity of the y direction
z : Parallel translation quantity of the z direction

Drawing example: cone1.wrl



1-4. Column

main program cylinder1.f
subroutine cylinder1(rd, h, r, g, b)

Description: This program draws a column that it gives value of the color and radius of the basal plane, the height.

Input variable:

rd : The radius of the basal plane

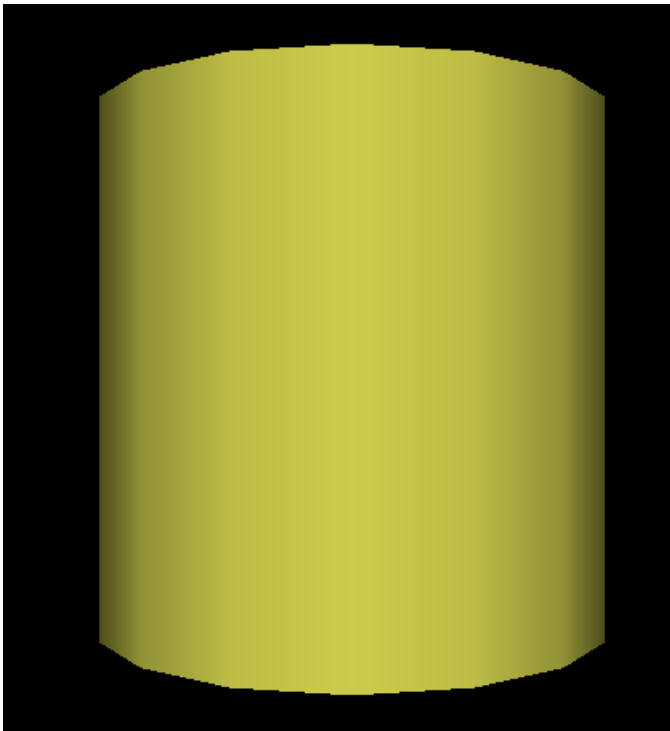
h : The height

r : The red value [0,1]

g : The green value [0,1]

b : The blue value [0,1]

Drawing example: cylinder1.wrl



1-5. Sphere

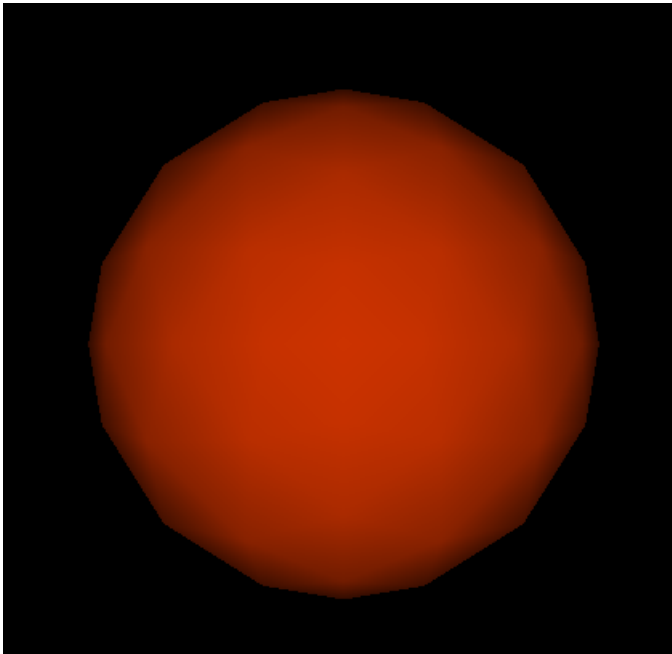
main program sphere1.f
subroutine shere2 (r, g, b, rd)

Description: This program draws a sphere that it gives value of the color and radius.

Input variable:

rd : The radius
r : The red value [0,1]
g : The green value [0,1]
b : The blue value [0,1]

Drawing example: sphere1.wrl



1-6. Background color

main program backgrd4.f
subroutine backd (n1, n2, rd)

Description: This program gives the color to the background.

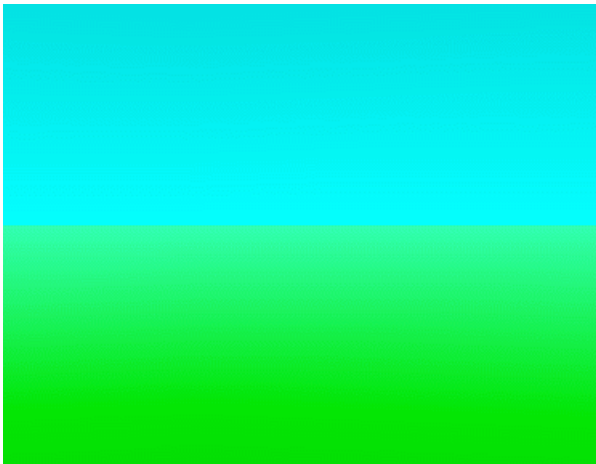
Input variable:

n1 : The number of the step in which the color of the ground changes

n2 : The number of the step in which the sky of the ground changes

rd : The angle per one step

Drawing example: backgrd4.wrl



1-7. Character string

main program text1.f
subroutine text1 (x, y, z, n, r, g, b, sz, ch1)

Description: This program draws character string that it gives character string and font color, font size.

Input variable:

x : Parallel translation quantity of the x direction
y : Parallel translation quantity of the y direction
z : Parallel translation quantity of the z direction
n : The number of the character string (maximum is 10)
r : The red value [0,1]
g : The green value [0,1]
b : The blue value [0,1]
sz : The font size
ch1 : The input character string [within 40 characters]

Drawing example: text1.wrl



Nagoya University
Solar-Terrestrial Environment Laboratory

1-8. Polygonal line : monochrome

main program lineset1.f
subroutine line1 (ln, po, r, g, b)

Description: This program draws a polygonal line in giving double end coordinate and color of the segment, and in connecting the coordinate in one stroke in the order.

Input variable:

ln : The data number.

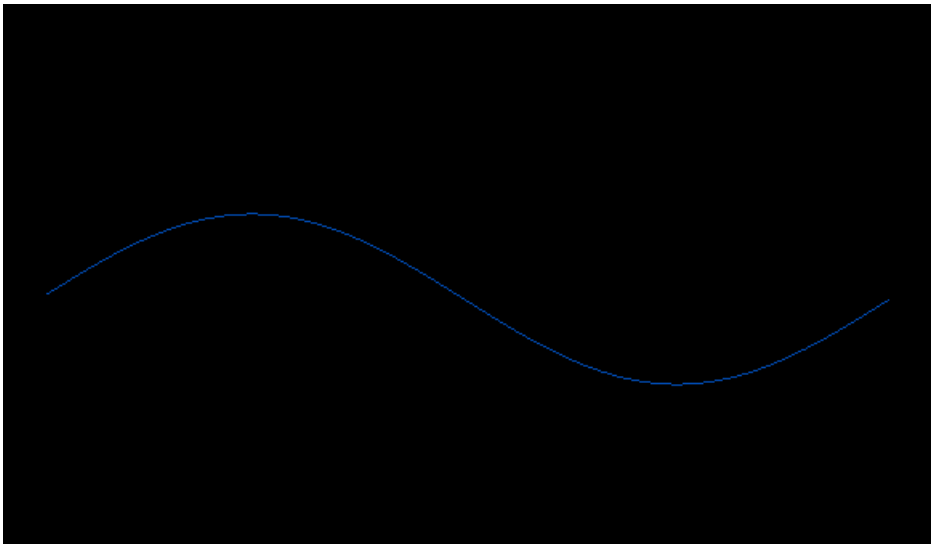
po(3*ln) : The coordinate of the point (x, y, z) is given in the order.

r : The red value. [0,1]

g : The green value. [0,1]

b : The blue value. [0,1]

Drawing example: lineset1.wrl



1-9. Polygonal line : gradation

main program lineset2.f
subroutine line2 (ln, po, sco, eco)

Description: This program draws a polygonal line in giving double end coordinate and color (gradation) of the segment, and in connecting the coordinate in one stroke in the order.

Input variable:

ln : The data number

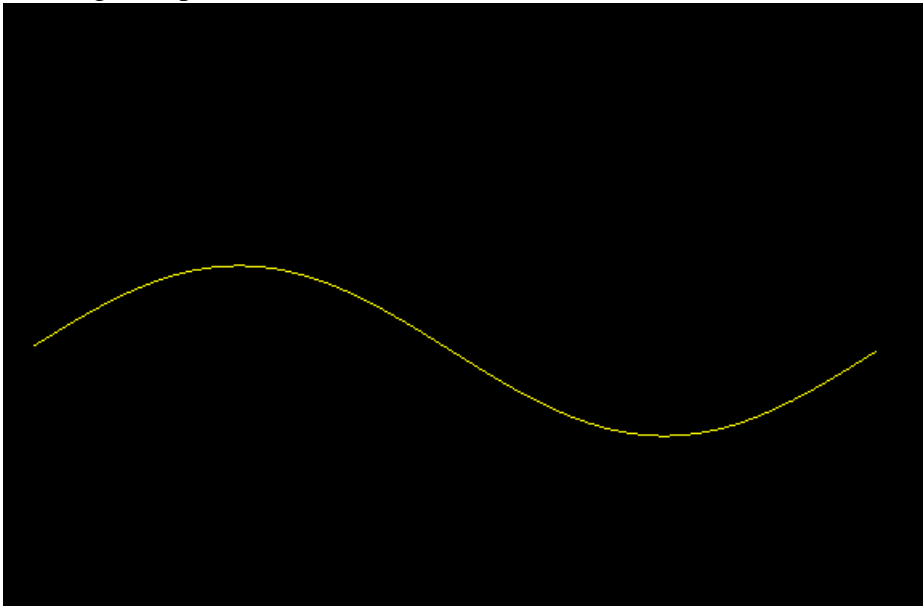
po(3*ln) : The coordinate of the point (x, y, z) is given in the order

sco : The color in the start of the gradation (color code)

eco : The color in the end of the gradation (color code)

color code (1: white 2: yellow 3: mazenda 4: red 5: sian 6: green 7: blue)

Drawing example: lineset2.wrl



1-10. Dotted line : monochrome

main program pointset1.f
subroutine point1 (ln, po, r, g, b)

Description: This program draws dotted line that it gives the coordinate of the point.

Input variable:

ln : The data number

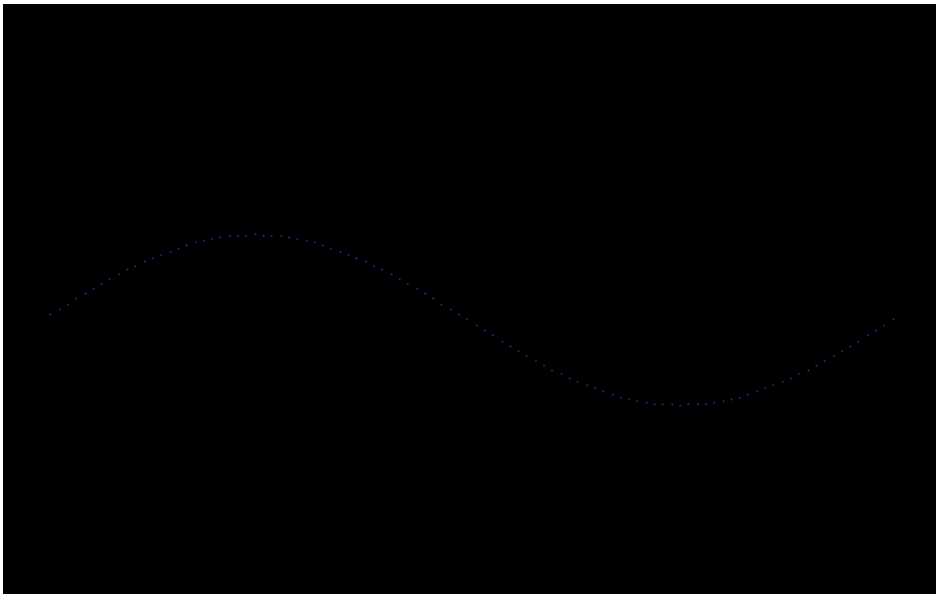
po(3*ln) : The coordinate of the point (x, y, z) is given in the order

r : The red value [0,1]

g : The green value [0,1]

b : The blue value [0,1]

Drawing example: pointset1.wrl



1-11. Dotted line : gradation

main program pointset2.f
subroutine point2 (ln, po, sco, eco)

Description: This program draws dotted line that it gives the coordinate of the point.

Input variable:

ln : The data number

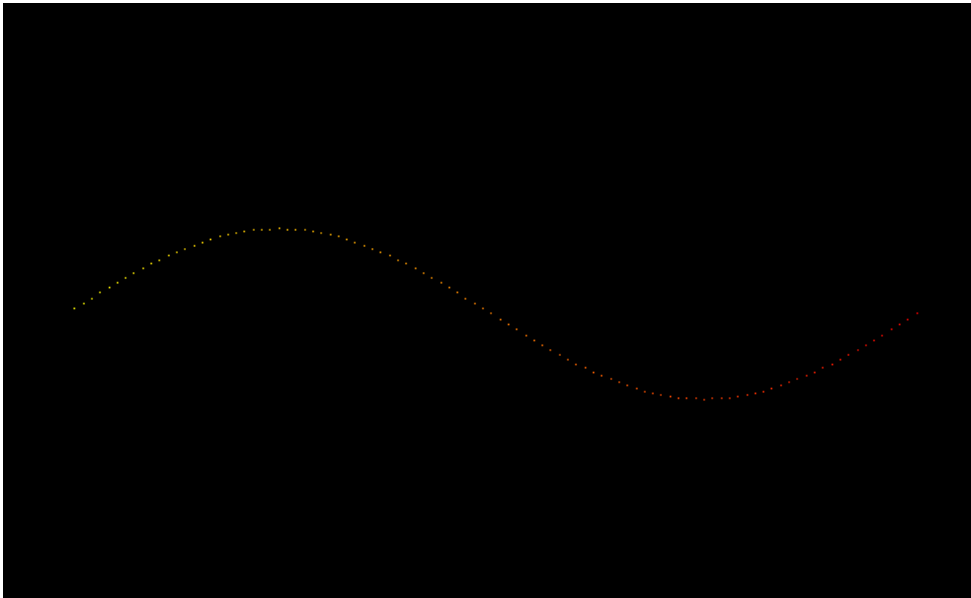
po(3*ln) : The coordinate of the point (x, y, z) is given in the order

sco : The color in the start of the gradation (color code)

eco : The color in the end of the gradation (color code)

color code (1: white 2: yellow 3: mazenda 4: red 5: sian 6: green 7: blue)

Drawing example: pointset2.wrl



1-12. Arrow with the thickness

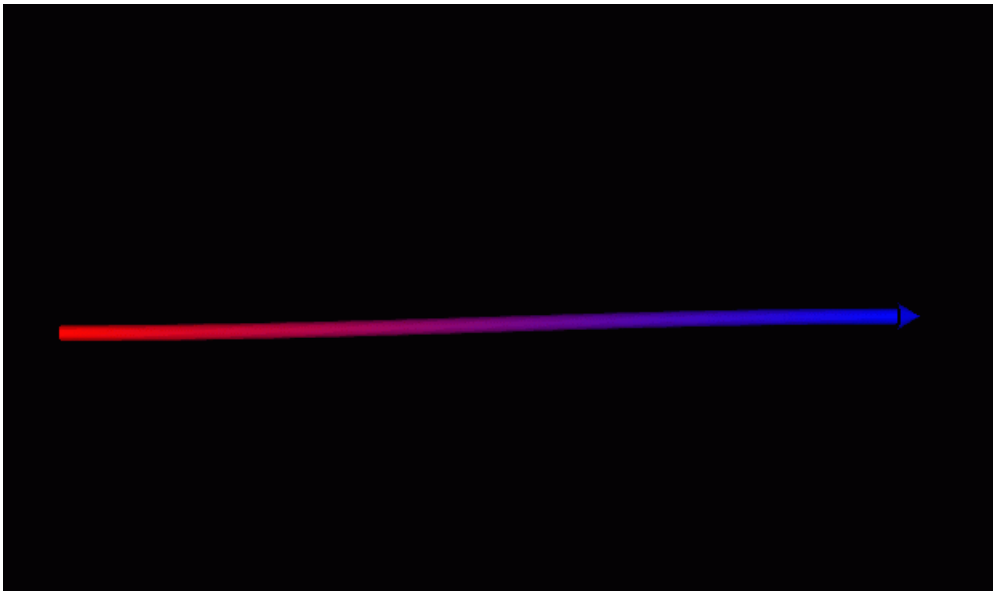
main program arrow2.f
subroutine arrow2 (nx, ny, u, p, rd1, rd2, h, cx, cy, cz, br, bg, bb, cr, cg, cb)

Description: This program draws an arrow with the thickness by combining the cylinder (gradation color) with the trigonal pyramid.

Input variable:

nx : Data number of the x direction
ny : Data number of the y direction
u : Color (r, g, b) of the point is given in the order
p : The coordinate of the point (x, y, z) is given in the order
rd1 : The radius of the cylindrical basal plane
rd2 : The radius of the basal plane of trigonal pyramid
h : The height of the trigonal pyramid
cx : The central x coordinate of the trigonal pyramid
cy : The central y coordinate of the trigonal pyramid
cz : The central z coordinate of the trigonal pyramid
br : The red value of the cylinder basal plane [0,1]
bg : The green value of the cylinder basal plane [0,1]
bb : The blue value of the cylinder basal plane [0,1]
cr : The red value of the trigonal pyramid [0,1]
cg : The green value of the trigonal pyramid [0,1]
cb : The blue value of the trigonal pyramid [0,1]

Drawing example: arrow2.wrl



1-13. Curve with the thickness

main program linemesh1.f
subroutine linemesh (p, n1, rd, r, g, b)

Description: This program draws a line with the thickness by connecting cylinder.

Input variable:

p : The coordinate of the point (x, y, z) is given in the order

n1 : The data number

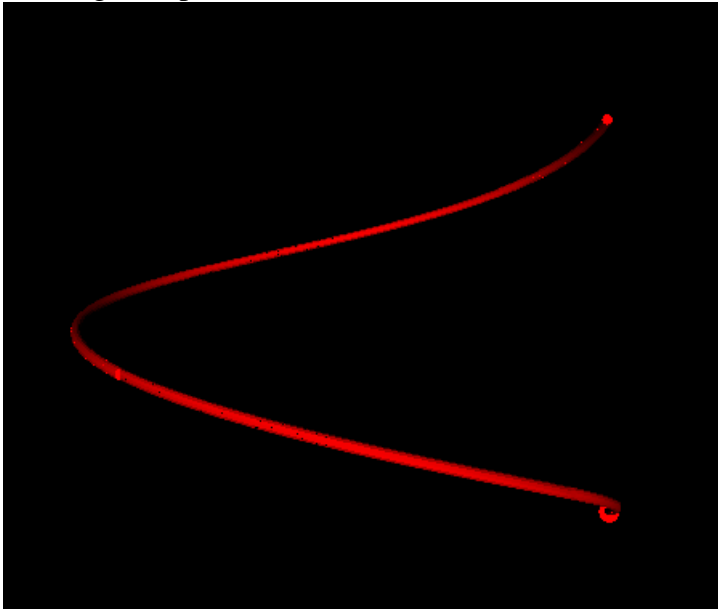
rd : The radius of the cylinder

r : The red value of the cylinder [0,1]

g : The green value of the cylinder [0,1]

b : The blue value of the cylinder [0,1]

Drawing example: linemesh1.wrl



1-14. Plane (1) : Monochromatic triangular mesh

main program mesh1.f
subroutine face1 (ln, po, r, g, b)

Description: This program draws a plane by triangular mesh that it gives the coordinate of the point and.

Input variable:

ln : The data number

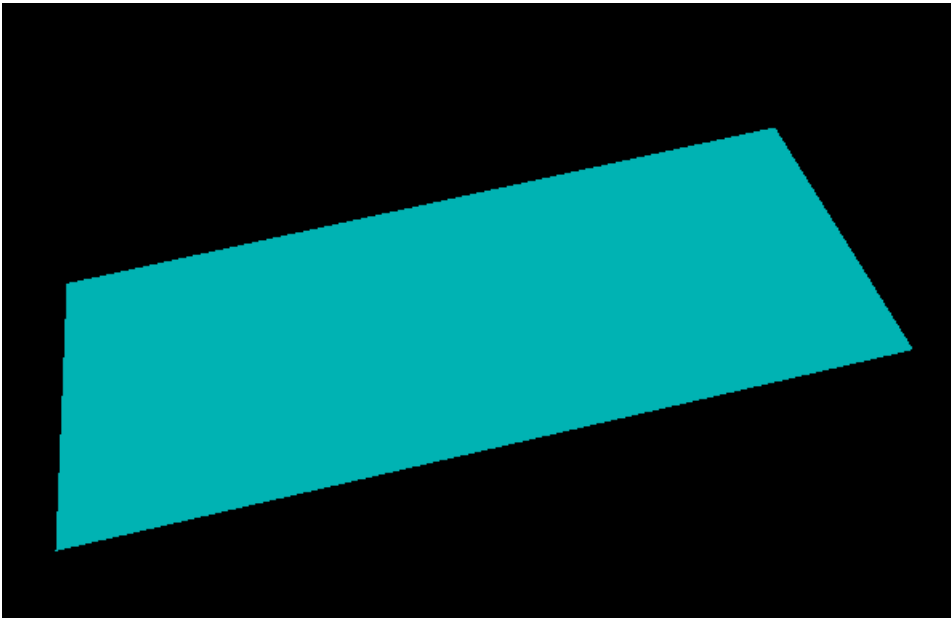
po : The coordinate of the point (x, y, z) is given in the order

r : The red value of the plane [0,1]

g : The green value of the plane [0,1]

b : The blue value of the plane [0,1]

Drawing example: mesh1.wrl



1-15. Plane (2) : Triangular belt

main program triangb1.f
subroutine triangbk1 (nx, ny, u, p)

Description: This program draws a plane by the triangular belt that it gives coordinate of point data and color.

Input variable:

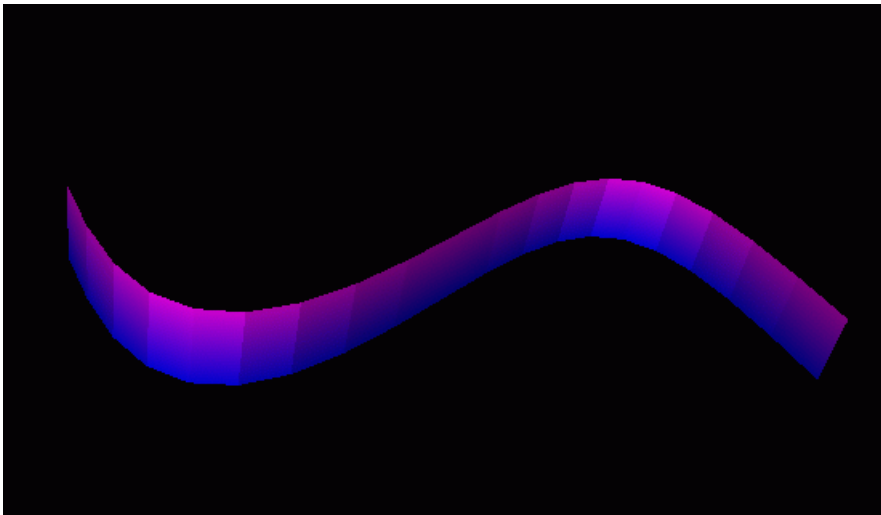
nx : Data number of the x direction

ny : Data number of the y direction

u : Color (r, g, b) of the point is given in the order

p : The coordinate of the point (x, y, z) is given in the order

Drawing example: triangb1.wrl



1-16. Plane (3) : Polychrome triangular mesh

main program `triangm1.f`
subroutine `triangm (nx, ny, u, p)`

Description: This program draws a plane by the triangular mesh that it gives coordinate of point data and color.

Input variable:

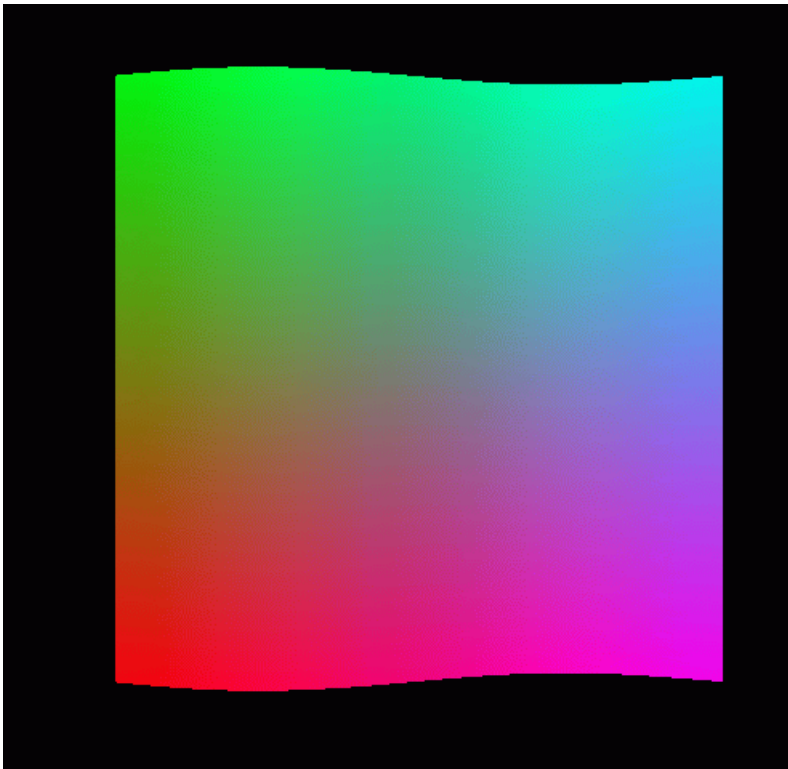
`nx` : Data number of the x direction

`ny` : Data number of the y direction

`u` : Color (r, g, b) of the point is given in the order

`p` : The coordinate of the point (x, y, z) is given in the order

Drawing example: `triangm1.wrl`



1-17. Solid : It is composed of the triangular mesh

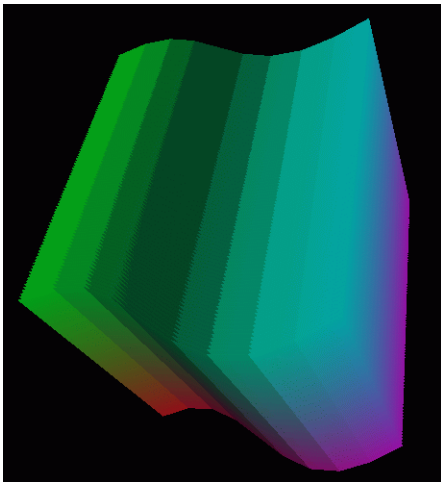
main program defusem1.f
subroutine defusem (nx, ny, u, p, x1, y1, z1, x2, y2, z2, r, n)

Description: This program draws a solid by the plane made from the data of the point by the triangular mesh gradually moving.

Input variable:

nx : Data number of the x direction
ny : Data number of the y direction
u : Color (r, g, b) of the point is given in the order
p : The coordinate of the point (x, y, z) is given in the order
x1 : Parallel translation quantity of the x direction
y1 : Parallel translation quantity of the y direction
z1 : Parallel translation quantity of the z direction
x2 : x coordinate of the rotating shaft
y2 : y coordinate of the rotating shaft
z2 : z coordinate of the rotating shaft
r : The transfer rotation angle (radian)
n : The processing frequency

Drawing example: defusem1.wrl



1-18. Texture mapping

main program image1.f
subroutine imagetx1 (ln, po, ch1, n)

Description: This program sticks a textured image designated for the plane drew by the triangular mesh.

Input variable:

ln : The data number.

po : The coordinate of the point (x, y, z) is given in the order.

ch1 : The URL of the textured image.

n : The number of characters of the URL.

Drawing example: image1.wrl



1-19. The complicated curved surface : Triangular mesh

main program ebmesh3d.f
subroutine triangm(nx, ny, u, p)

Description: This program is draws a picture by the triangular mesh it reads the point data of constituting the plane.

Input variable:

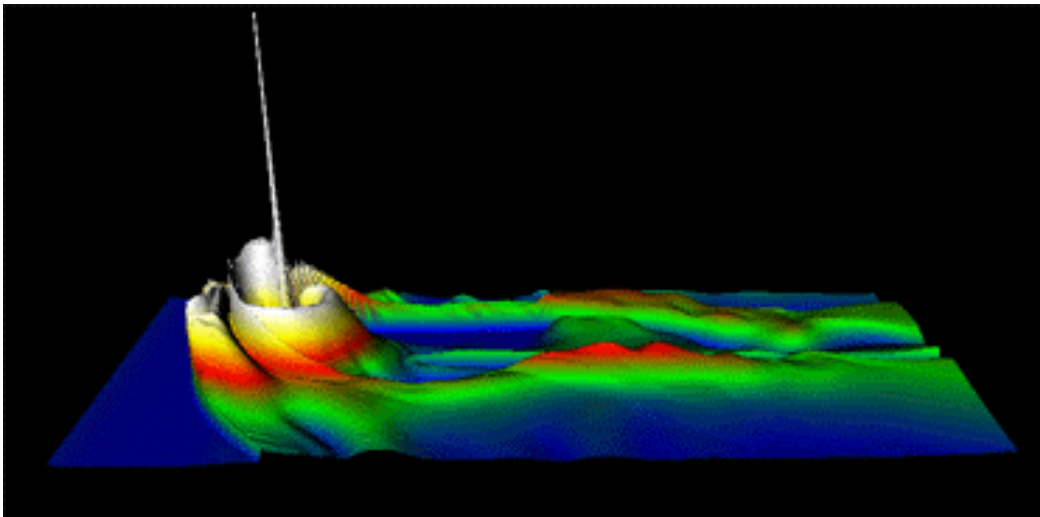
nx : Data number of the x direction

ny : Data number of the y direction

u : Color (r, g, b) of the point is given in the order

p : The coordinate of the point (x, y, z) is given in the order

Drawing example: ebmesh3d.wrl



2. Fundamental example (2)

In this chapter, the example that it is fundamental for making the image file of the VRML 2.0 type using the Fortran program and can utilize as it is for analysis application of simulation data is shown. Concretely, the image file of the VRML2.0 type is made by compiling, linking and executing each main program and subroutine package (zvrsuba.f).

The usage:

1. `f77 -c -O zvrsuba.f`

The object module (zvrsuba.o) is made by the compilation of subroutine package (zvrsuba.f).

2. `f77 -O msymbol.f zvrsuba.o`

The execution file (a.out) is made by the link of the main program (msymbol.f etc.) with the object module (zvrsuba.o).

3. `a.out`

The VRML 2.0 (fort.10) file is made by the execution of the execution file (a.out).

4. `mv fort.10 box1.wrl`

The file name of made VRML is changed at "****.wrl".

2-1. Character

main program msymbol.f

subroutine symblv (x, y, z, h, chr, n) ...white

symbmvc (x, y, z, h, r, g, b, chr, n) ...multicolor

Description: This program draws characters by giving width and height (, color) of character and origin, character.

Input variable:

x : The x coordinate of the origin

y : The y coordinate of the origin

z : The z coordinate of the origin

h : The height of the character

r : The red value [0,1]

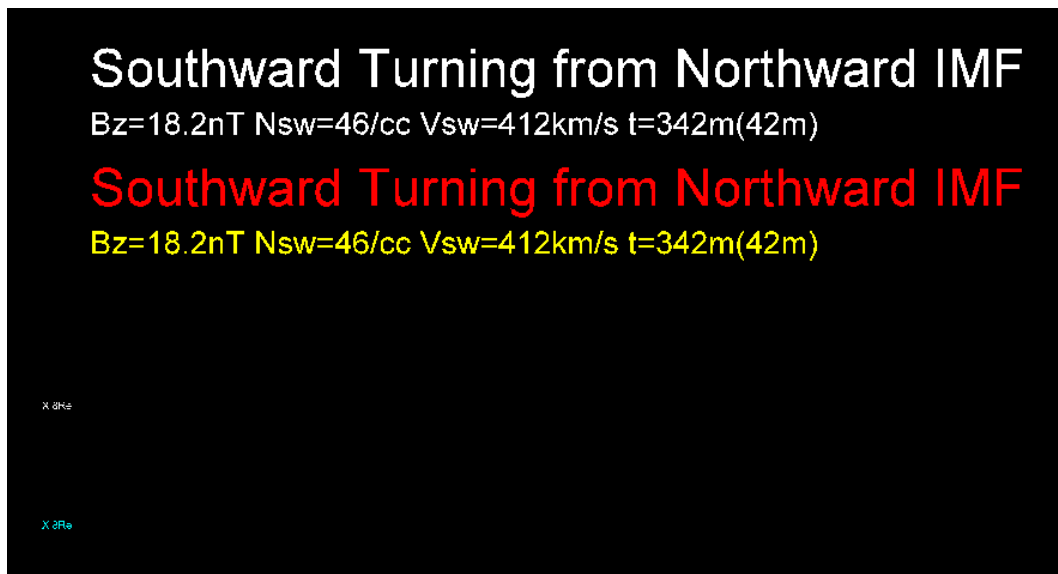
g : The green value [0,1]

b : The blue value [0,1]

chr : The input character [within 80 characters]

n : The number of the character

Drawing example: msymbol.wrl



2-2. Point

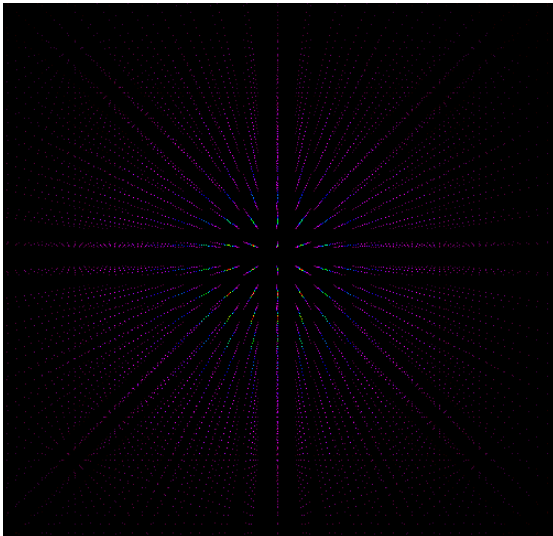
```
main program  mpoint.f
subroutine  point3d (nx, ny, nz, x0, y0, z0, xl, yl, zl, ico, u)
```

Description: This program draws points by giving position and colored code of the three-dimensional point of the $nx*ny*nz$ piece.

Input variable:

- nx : Data number of the x direction
- ny : Data number of the y direction
- nz : Data number of the z direction
- x0 : The x coordinate of the origin
- y0 : The y coordinate of the origin
- z0 : The z coordinate of the origin
- x1 : The length of the x direction
- y1 : The length of the y direction
- z1 : The length of the z direction
- ico : color code [1,7]
- u ($nx*ny*nz$) : They are input data, and the coordinate of the multipoint (x, y, z) is given in the order

Drawing example: mpoint.wrl



2-3. Line

```
main program  mline.f  
subroutine  line3d (lasl, r, g, b, po)
```

Description: This program draws a line in giving coordinate and color of sequence of points, and in connecting sequence of points in the one stroke.

Input variable:

lasl : The number of sequence of points

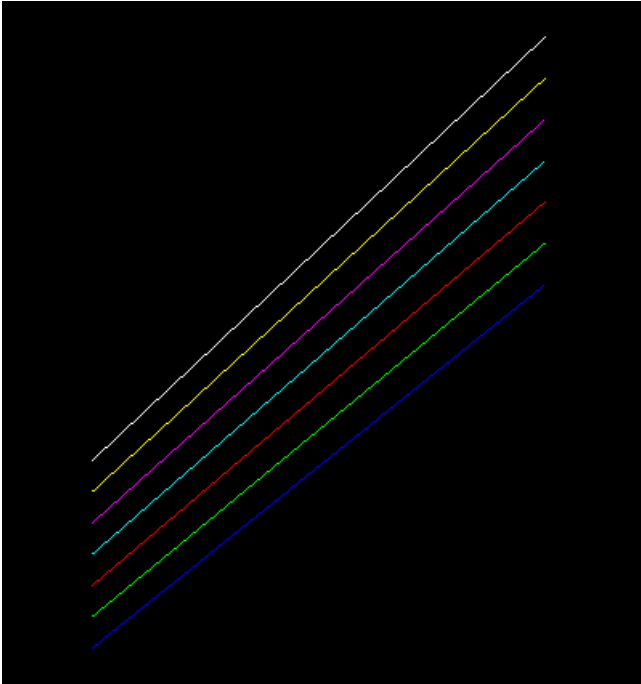
r : The red value [0,1]

g : The green value [0,1]

b : The blue value [0,1]

po (3*lasl) : The coordinate of sequence of points (x, y, z) is given in the order

Drawing example: mline.wrl



2-4. Triangular mesh

```
main program  mtriangm.f  
subroutine  triangm (nx, ny, u, p)
```

Description: This program draws the image of the two-dimensional data of the coordinate of the $nx*ny$ piece by giving position in the origin and length of one side, data number, colored code by the triangular mesh.

Input variable:

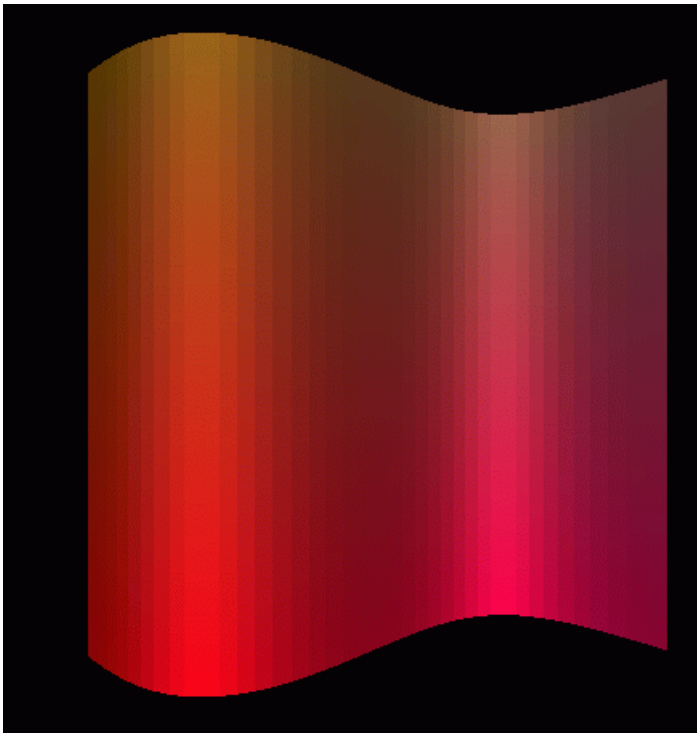
nx : Data number of the x direction

ny : Data number of the y direction

u ($3*nx*ny$) : Input data

p ($3*nx*ny$) : Auxiliary data necessary for the calculation

Drawing example: mtriangm.wrl



2-5. Pixel image

main program mzpt03.f

subroutine pixel2(nx, ny, xb, yb, xl, yl, ipx0, ico, icc, zcc, u)

Description: This program draws a image of the two-dimensional data u ($nx*ny$) by giving position in the origin and length of one side, data number, colored code in the pixel image.

Input variable:

nx : Data number of the x direction

ny : Data number of the y direction

xb : The x coordinate of the origin

yb : The y coordinate of the origin

xl : The length of the x direction

yl : The length of the y direction

ipx0 : 3=rgb color, 4=rgb color and transparency can be set

ico : color code [1,7]

icc : Picture is drawn in the 1=xy plane, 2=xz plane, 3=yz plane

zcc : The position in chosen plane

u ($nx*ny$) : Input data

Drawing example: mzpt03.wrl



2-6. Frame and pixel image and character

main program mpix015.f

Description: This program draws picture by combining frame and pixel image and character.

Input variable: x, y, z, h, ch1, n, icc, zcc, r, g, b, x1, y1, x2, y2, nx, ny, xb, yb, x1, y1, ipx0, ico, u
It is well-informed the each subroutine reference.

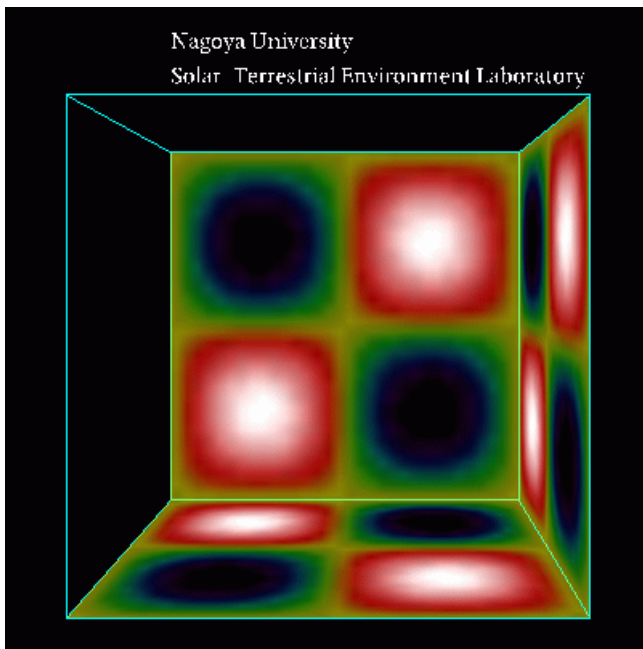
Using subroutine :

symblv (x, y, z, h, ch1, n) :

rect1 (icc, zcc, r, g, b, x1, y1, x2, y2) :

pixel2 (nx, ny, xb, yb, x1, y1, ipx0, ico, icc, zcc, u) :

Drawing example: mpix015.wrl



2-7. Equivalence plane

main program mcube301.f

subroutine csuba (ic, aa, u) < (nx, ny, nz, x0, y0, z0, xl, yl, zl, nol, ico)

Description: This program draws equivalence plane by giving and three-dimensional data (u (nx*ny*nz)).

Input variable:

nx = ic (2) : Data number of the x direction

ny = ic (3) : Data number of the y direction

nz = ic (4) : Data number of the z direction

x0 = aa (11) : The x coordinate of the origin

y0 = aa (12) : The y coordinate of the origin

z0 = aa (13) : The z coordinate of the origin

xl = aa (8) : The length of the x direction

yl = aa (9) : The length of the x direction

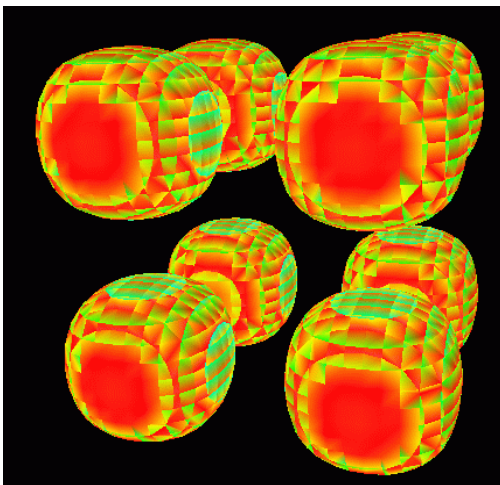
zl = aa (10) : The length of the x direction

ico = ic (1) : color code [1,7]

nol : The number of the edge of the rectangular parallelepiped ;12

u (nx*ny*nz) : Three-dimensional input data

Drawing example: mcube301.wrl



2-8. Outer magnetosphere by three-dimensional lattice

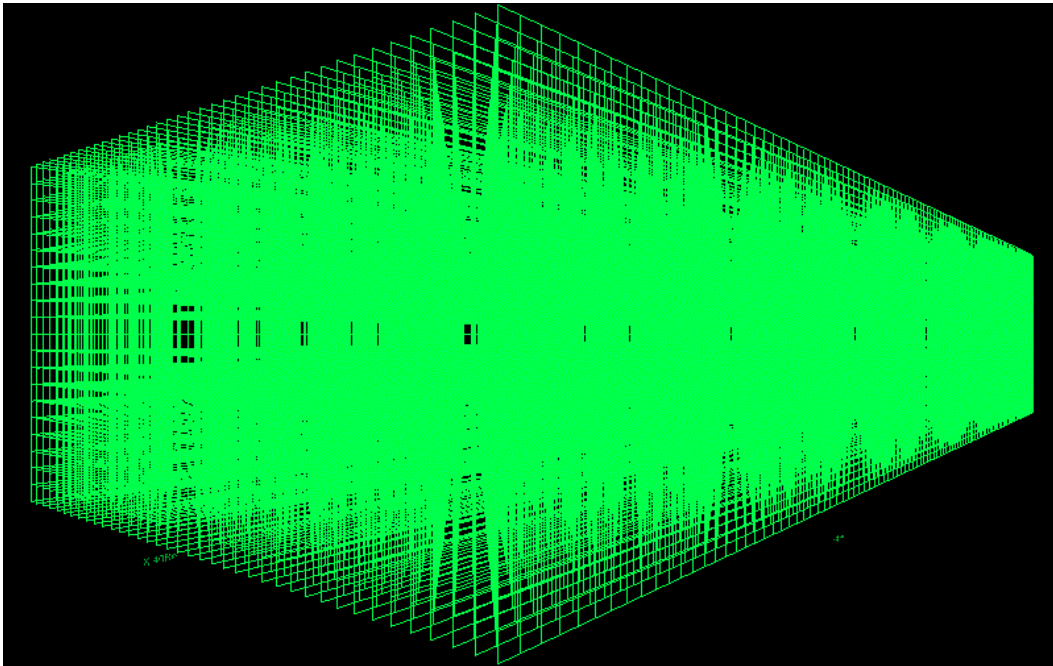
```
main program  outline3d.f  
subroutine  lat3d1 ( ic, aa ) < ( nx, ny, nz, xl, yl, zl, r, g, b )
```

Description: This program draws the outer magnetosphere by three-dimensional lattice.

Input variable:

$nx = ic(1)$: Data number of the x direction
 $ny = ic(2)$: Data number of the y direction
 $nz = ic(3)$: Data number of the z direction
 $xl = aa(1)$: The length of the x direction
 $yl = aa(2)$: The length of the x direction
 $zl = aa(3)$: The length of the x direction
 $r = aa(4)$: The red value [0,1]
 $g = aa(5)$: The green value [0,1]
 $b = aa(6)$: The blue value [0,1]

Drawing example: outline3d.wrl



2-9. Inner magnetosphere by three-dimensional lattice

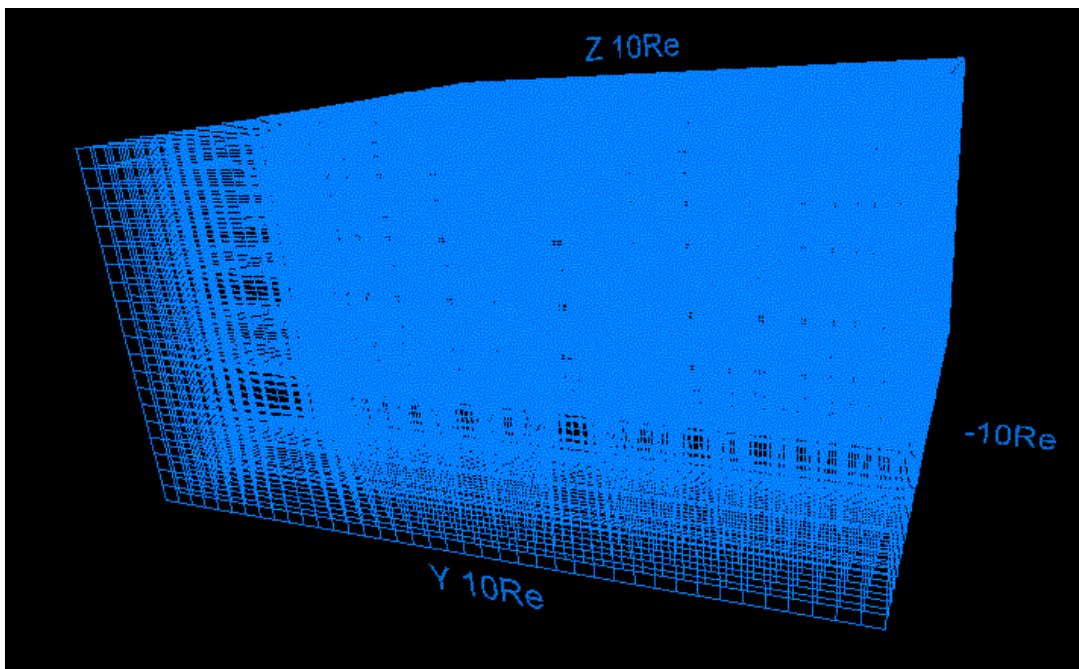
```
main program  outline3d.f
subroutine  lat3d2 ( ic, aa ) < (nx, ny, nz, xl, yl, zl, r, g, b)
```

Description: This program draws the inner magnetosphere by three-dimensional lattice.

Input variable:

$nx = ic(1)$: Data number of the x direction
 $ny = ic(2)$: Data number of the y direction
 $nz = ic(3)$: Data number of the z direction
 $xl = aa(1)$: The length of the x direction
 $yl = aa(2)$: The length of the x direction
 $zl = aa(3)$: The length of the x direction
 $r = aa(4)$: The red value [0,1]
 $g = aa(5)$: The green value [0,1]
 $b = aa(6)$: The blue value [0,1]

Drawing example: inline3d.wrl



2-10. Ionosphere by three-dimensional lattice

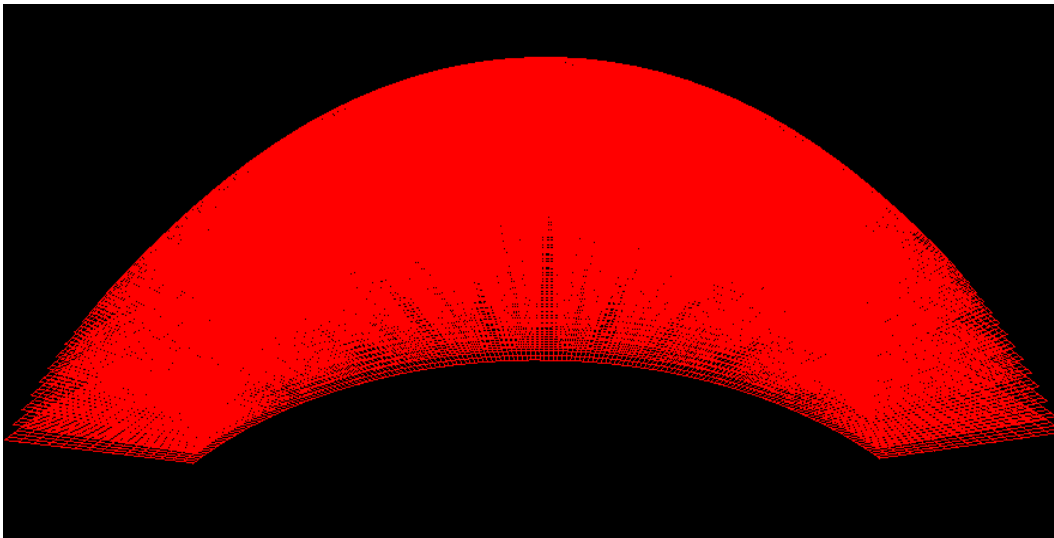
```
main program  denline3d.f  
subroutine  lat3d3 ( ic, aa ) < ( re1, re2, r, g, b )
```

Description: This program draws the ionosphere by three-dimensional lattice.

Input variable:

re1 = aa(1) : The radius of inner circle
re2 = aa(2) : The radius of outer circle
r = aa(3) : The red value [0,1]
g = aa(4) : The green value [0,1]
b = aa(5) : The blue value [0,1]

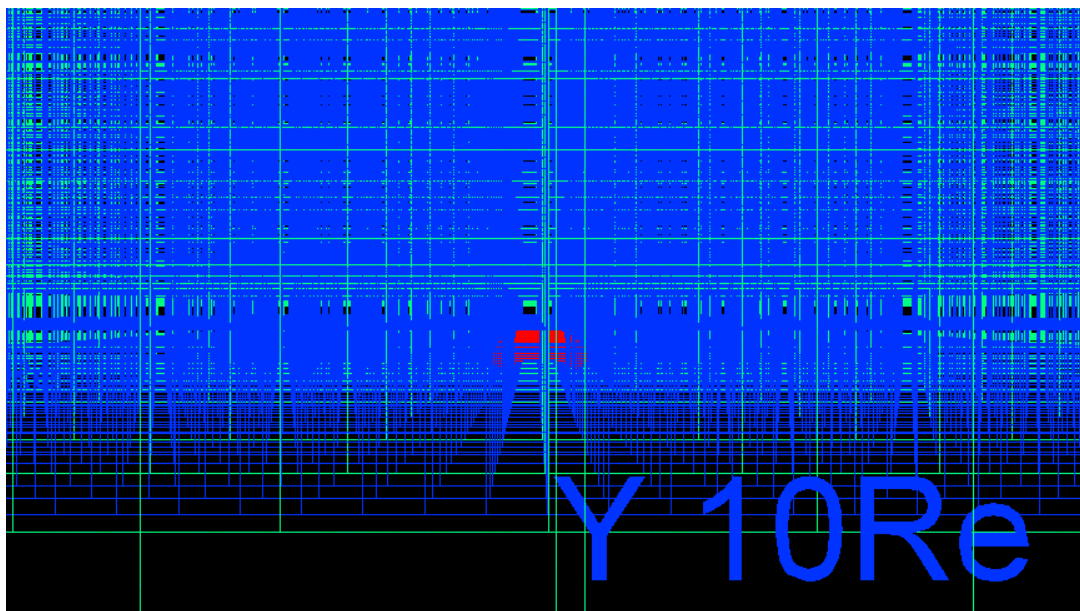
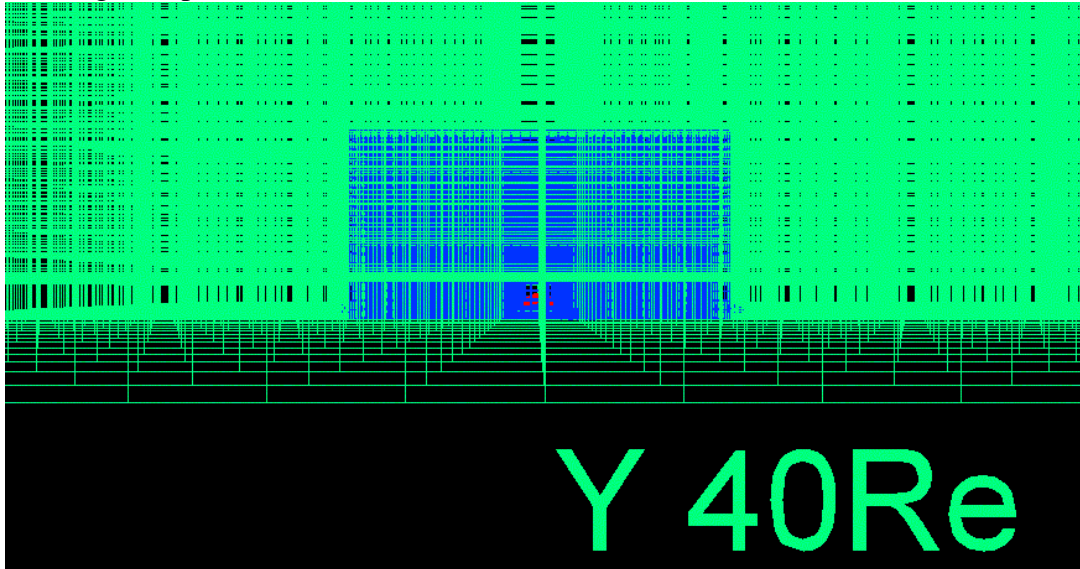
Drawing example: denline3d.wrl



2-11. Unification of outer / inner magnetosphere and inosphere

```
cat outline3d.wrl inline3d.wrl denline3d.wrl > line3d.wrl
```

Drawing example: line3d.wrl



3. Application to the MHD simulation

In this chapter, the concrete application which makes the VRML file by using the asymmetrical three-dimensional magnetosphere data of the three-dimensional global MHD simulation of interaction between the solar wind and the earth's magnetosphere with dawn-dusk asymmetry, is shown.

The usage:

1. `f77 -c -O zvrsuba.f`

The object module (zvrsuba.f) is made by the compilation of subroutine package (zvrsuba.o).

2. `f77 -O zvrcroa.f zvrsuba.o`

The execution file (a.out) is made by the link of the main program (zvrcroa.f or zvmaga.f) with the object module (zvrsuba.o).

3. a.out

The VRML 2.0 (fort.10) file is made by the execution of the execution file (a.out).

4. `mv fort.10 box1.wrl`

The file name of made VRML is changed at "***.wrl".

5. `cat zvrcroa.wrl zvmaga.wrl > zvr1.wrl`

The file of two VRML is synthesized.

3-1. Sectional views such as plasma temperature of the earth's magnetosphere are drawn by the pixel image

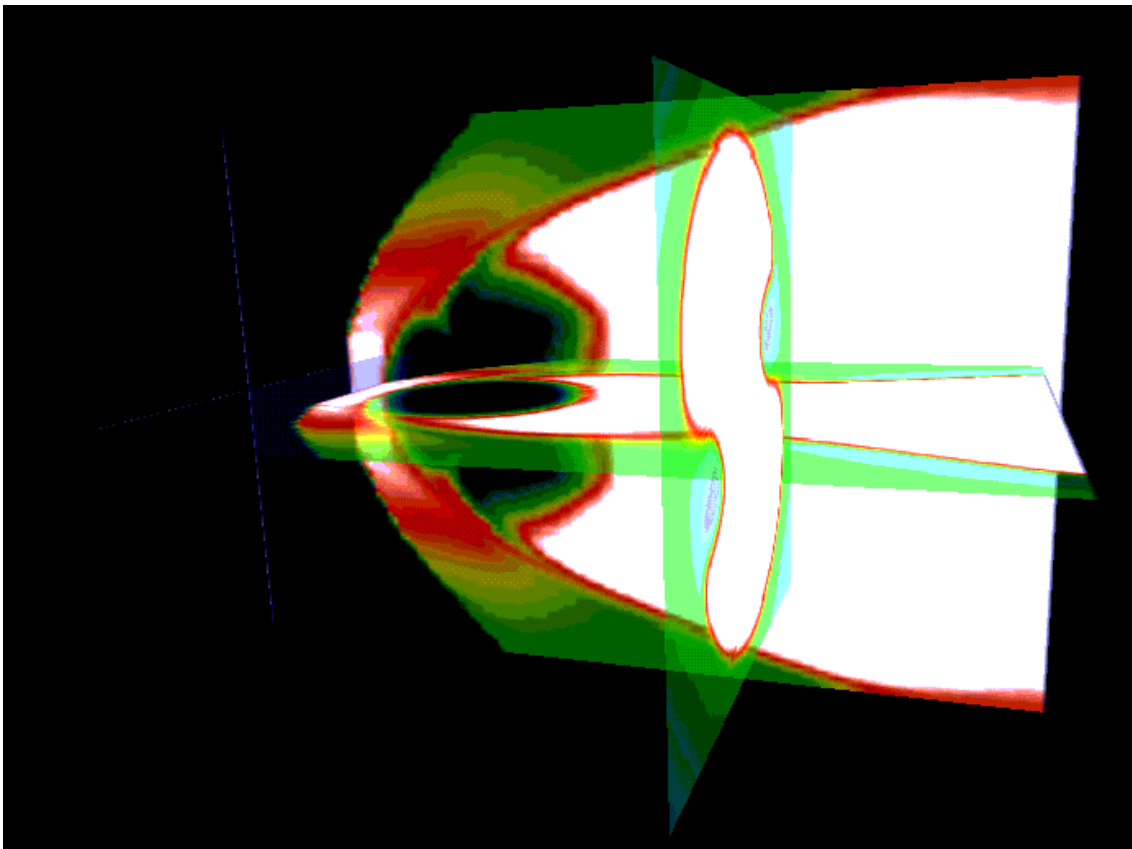
VRML file which describes sectional views such as plasma temperature of the earth's magnetosphere in the pixel image is made to the three-dimensional MHD simulation data with dawn-dusk asymmetry, in executing main program: zvrcoa.f and subroutine package: zvrsuba.f, is made.

Main using subroutine:

```
pixel1(nx,ny,xb,yb,xl,yl,ipx0,ico,icc,zcc,vmin,vmax,u)
```

3D MHD simulation code which receives the result. : eartha10.f

Drawing example: This drawing example is drawn pixel image of the sectional view of the three-dimensional MHD simulation result of the earth's magnetosphere using the translucency method. : zvrcoa.wrl



The details of main subroutine.

<The pixel image using maximum value and minimum value>

```
subroutine pixel1(nx,ny,xb,yb,xl,yl,ipx0,ico,icc,zcc,vmin,vmax,u)
```

Description: This program draws pixel image using maximum value (vmax) and minimum value (vmin) of $u(i1)$ are given from the outside, $u(i1)$ is normalized, and by giving position in the origin and length of one side, data number, colored code, the image of two-dimensional data

$u(nx*ny)$.

Input variable:

nx : Data number of the x direction

ny : Data number of the y direction

xb : The x coordinate of the origin

yb : The y coordinate of the origin

xl : The length of the x direction

yl : The length of the y direction

$ipx0$: 3=rgb color, 4=rgb color and transparency can be set

ico : color code [1,7]

icc : Picture is drawn in the 1=xy plane, 2=xz plane, 3=yz plane

zcc : The position in chosen plane

$u(nx*ny)$: Input data

$vmin$: The maximum value of $u(i1)$ given from the outside

$vmax$: The minimum value of $u(i1)$ given from the outside

$u(nx*ny)$: Input data $u(i1)$, $i1=i+nx*(j-1)$

Main using subroutine:

$pixebe(nx,ny,ipx0)$: The start of the pixel image

$pixeco(nx,ny,ico,u)$: The setting of the rgb color

$pixect(nx,ny,ico,u)$: The setting of transparency and rgb color

$pixeen(xb,yb,xl,yl,icc,zcc)$: The end of the pixel image

3-2. Three-dimensional structure of the earth's magnetosphere

This program draws the three-dimensional structure of line of magnetic force of the earth's magnetosphere in making to be inputting three-dimensional MHD simulation data with dawn-dusk asymmetry, in executing main program: zvmaga.f and subroutine package: zvrsuba.f.

Main using subroutine:

ainte1a(ia,aa,f,p)

zsub33(ia,aa)

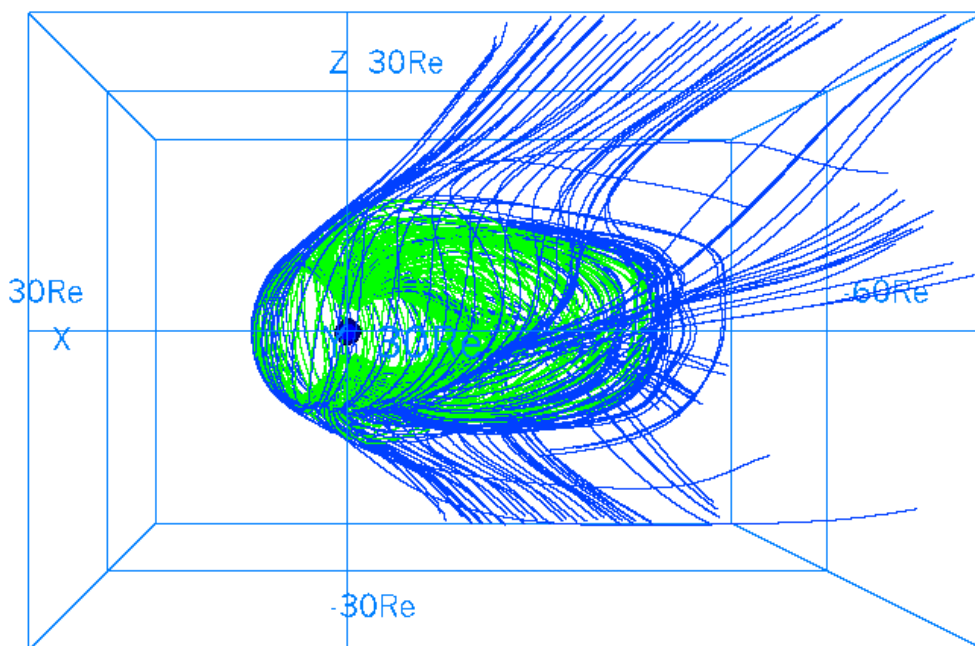
ainte1a(ia,aa,f,p)

ainte21(ia,aa,f,p)

MHD simulation code which receives the result.: eartha10.f

Drawing example: zvmaga.wrl

T = 22-10-1999 06:30:00



Details of main subroutine

<The decision by beforehand describing line of magnetic force>

subroutine ainte1a(ia,aa,f,p)

Description: Line of magnetic force is tracked by using magnetic field data of 3 components from three-dimensional MHD simulation data f(i1), closed, detached, line of magnetic force is judged.

Input variable:

nx=ia(1) : Data number of the x direction in the starting point of line of magnetic force

ny=ia(2) : Data number of the y direction in the starting point of line of magnetic force

mx=ia(3) : Data number of the x direction of three-dimensional data f(i1)
 my=ia(4) : Data number of the y direction of three-dimensional data f(i1)
 mz=ia(5) : Data number of the z direction of three-dimensional data f(i1)
 nxz=ia(6) : The parameter which adjusts the global center
 mi=ia(7) : The first position of the input variable
 mo=ia(8) : The first position of the output variable
 th0=aa(1)*pi/180.0 : The lowest latitude in the starting point of line of magnetic force
 aru=aa(2) : The parameter which characterizes the transition of the inner edge
 arl=aa(3) : The outside edge radius of the transition region of inner edge
 arb=aa(4) : The edge radius within the transition region of inner edge
 xxl=aa(5) : The length of simulation box of the x direction
 yyl=aa(6) : The length of simulation box of the y direction
 zzl=aa(7) : The length of simulation box of the z direction
 b0=aa(8) : The size of the dipole magnetic field (generally 1.0)
 gx0=aa(13) : The x coordinate of the origin which describes the graph
 gy0=aa(14) : The y coordinate of the origin which describes the graph
 gxl=aa(15) : The length of x direction which describes the graph
 gyl=aa(16) : The length of y direction which describes the graph
 gth=aa(17)*pi/180.0 : The rotation angle of the coordinate of the graph
 gxmi=aa(18) : The minimum value of x direction which describes the graph
 gxma=aa(19) : The maximum value of x direction which describes the graph
 ep1=aa(20) : The minimum threshold of decision in length of field line

Main using subroutine:

quant1(x,y,z,hx,hy,hz,l,mx,my,mz,nxp,aa,f,q) :
 MHD data q(8) of optional place (x,y,z) is calculated from MHD data f(i1)
 of the three-dimensional lattice point.

<Three-dimensional line of magnetic force starting from polar region>

subroutine aintel1(ia,aa,f,p)

Description: This program draws three-dimensional line of magnetic force starting from polar region using magnetic field data of 3 components from three-dimensional MHD simulation data f(i1).

Input variable:

nx=ia(1) : Data number of the x direction in the starting point of line of magnetic force
 ny=ia(2) : Data number of the y direction in the starting point of line of magnetic force
 mx=ia(3) : Data number of the x direction of three-dimensional data f(i1)
 my=ia(4) : Data number of the y direction of three-dimensional data f(i1)
 mz=ia(5) : Data number of the z direction of three-dimensional data f(i1)
 nxz=ia(6) : The parameter which adjusts the global center
 mi=ia(7) : The first position of the input variable
 mo=ia(8) : The first position of the output variable
 th0=aa(1)*pi/180.0 : The lowest latitude in the starting point of line of magnetic force
 aru=aa(2) : The parameter which characterizes the transition of the inner edge
 arl=aa(3) : The outside edge radius of the transition region of inner edge
 arb=aa(4) : The edge radius within the transition region of inner edge

xxl=aa(5) : The length of simulation box of the x direction
 yyl=aa(6) : The length of simulation box of the y direction
 zzl=aa(7) : The length of simulation box of the z direction
 b0=aa(8) : The size of the dipole magnetic field (generally 1.0)
 gx0=aa(13) : The x coordinate of the origin which describes the graph
 gy0=aa(14) : The y coordinate of the origin which describes the graph
 gxl=aa(15) : The length of x direction which describes the graph
 gyl=aa(16) : The length of y direction which describes the graph
 gth=aa(17)*pi/180.0 : The rotation angle of the coordinate of the graph
 gxmi=aa(18) : The minimum value of x direction which describes the graph
 gxma=aa(19) : The maximum value of x direction which describes the graph
 ep1=aa(20) : The minimum threshold of decision in length of field line

Main using subroutine:

quant1(x,y,z,hx,hy,hz,1,mx,my,mz,nxp,aa,f,q) :
 MHD data q(8) of optional place (x,y,z) is calculated from MHD data f(i1)
 of the three-dimensional lattice point.
 linebe(r,g,b) : The start of the polygonal line description
 linep2(x1,y2,z1,1,lasl) : The execution of the polygonal line description
 lineen(lasl) : The end of the polygonal line description

<Three-dimensional line of magnetic force starting from equatorial plane>
 subroutine ainte21(ia,aa,f,p)

Description: This program draws three-dimensional line of magnetic force starting from equatorial plane using magnetic field data of 3 components from three-dimensional MHD simulation data f(i1).

Input variable:

nx=ia(1) : Data number of the x direction in the starting point of line of magnetic force
 ny=ia(2) : Data number of the y direction in the starting point of line of magnetic force
 mx=ia(3) : Data number of the x direction of three-dimensional data f(i1)
 my=ia(4) : Data number of the y direction of three-dimensional data f(i1)
 mz=ia(5) : Data number of the z direction of three-dimensional data f(i1)
 nxz=ia(6) : The parameter which adjusts the global center
 mi=ia(7) : The first position of the input variable
 mo=ia(8) : The first position of the output variable
 th0=aa(1)*pi/180.0 : The lowest latitude in the starting point of line of magnetic force
 aru=aa(2) : The parameter which characterizes the transition of the inner edge
 ar1=aa(3) : The outside edge radius of the transition region of inner edge
 arb=aa(4) : The edge radius within the transition region of inner edge
 xxl=aa(5) : The length of simulation box of the x direction
 yyl=aa(6) : The length of simulation box of the y direction
 zzl=aa(7) : The length of simulation box of the z direction
 b0=aa(8) : The size of the dipole magnetic field (generally 1.0)
 gx0=aa(13) : The x coordinate of the origin which describes the graph
 gy0=aa(14) : The y coordinate of the origin which describes the graph
 gxl=aa(15) : The length of x direction which describes the graph

gyl=aa(16) : The length of y direction which describes the graph
 gth=aa(17)*pi/180.0 : The rotation angle of the coordinate of the graph
 gxmi=aa(18) : The minimum value of x direction which describes the graph
 gxma=aa(19) : The maximum value of x direction which describes the graph
 ep1=aa(20) : The minimum threshold of decision in length of field line
 xlim=aa(23) : The restriction of the minimum value of the x coordinate
 xli2=aa(24) : The restriction of the minimum value of the y,z coordinate

Main using subroutine:

quant1(x,y,z,hx,hy,hz,1,mx,my,mz,nxp,aa,f,q) :
 MHD data q(8) of optional place (x,y,z) is calculated from MHD data f(i1)
 of the three-dimensional lattice point
 linebe(r,g,b) : The start of the polygonal line description
 linep2(x1,y2,z1,1,lasl) : The execution of the polygonal line description
 lineen(lasl) : The end of the polygonal line description

<The test of the boundary in the closed region with that it opens>

subroutine zsub33(ia,aa)

Description: The test result is handed in ainte1 from three-dimensional line of magnetic force of the earth's magnetosphere, in order to use with ainte1a result, and by selecting line of magnetic force which exists in the narrow region between open open region and closed region, in order to describe only the chosen line of magnetic force.

Input variable:

nxg=ia(1) : Data number of the x direction in the starting point of line of magnetic force
 nyg=ia(2) : Data number of the x direction in the starting point of line of magnetic force

Relating subroutine:

ainte1a(ia,aa,f,p)
 ainte1(ia,aa,f,p)

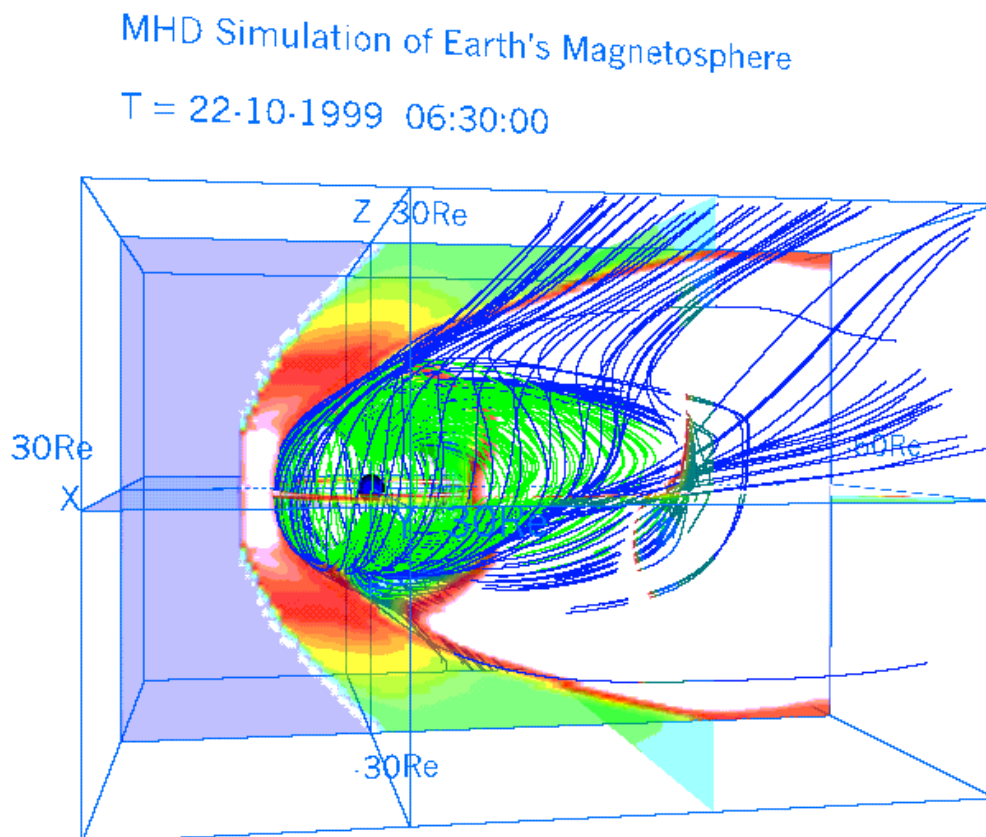
3-3. The synthesis of the three-dimensional image of pixel image and line of magnetic force

The example of synthesizing result of describing sectional views such as plasma temperature of the earth's magnetosphere in the pixel image (it refers to 3-1) and result of describing the three-dimensional structure of line of magnetic force of the earth's magnetosphere (it refers to 3-2) using "cat" is shown.

```
zvrcoa.wrl + zvmaga.wrl    zvr01.wrl
```

```
cat zvrcoa.wrl zvmaga.wrl > zvr01.wrl
```

Drawing example:zvr01.wrl



3-4. Three dimensional distribution of the plasma temperature in the whole earth's magnetosphere drawn by the multiple pixel image planes

The VRML file, which shows the three dimensional distribution of the plasma temperature in the whole earth's magnetosphere with dawn-dusk asymmetry by the multiple pixel image planes, is made by the three-dimensional MHD simulation data. It is executed by main program: zvrmpxa.f and subroutine package: zvrsuba.f.

Input variable:

tra : Transparency of pixel image plane (0.0 ~ 0.5 : 0.0 means 100% Transparency
0.5 means 50% Transparency)
ipn : Number of planes
icc : Direction of planes (1 : zx plane, 2 : yz plane, 3 : xy plane)

Main using subroutine (refer to 3-1):

pixel1(nx,ny,xb,yb,xl,yl,ipx0,ico,icc,zcc,vmin,vmax,tra,u)

Drawing example: This example shows how to draw the plasma temperature distribution of the earth's magnetosphere from the result of three-dimensional MHD simulation. It is drawn with multiple pixel image planes by using transparent method.

fig.3-1 : fa410zx.wrl	Example by drawing multiple zx planes (in case of icc=1 and ipn=11)
fig.3-2 : fa410yz.wrl	Example by drawing multiple yz planes (in case of icc=2, ipn=11)
fig.3-3 : fa410xy.wrl	Example by drawing multiple xy planes (in case of icc=3, ipn=11)
fig.3-4 : fa410xyz.wrl	a synthesis file with above three files by using "cat"

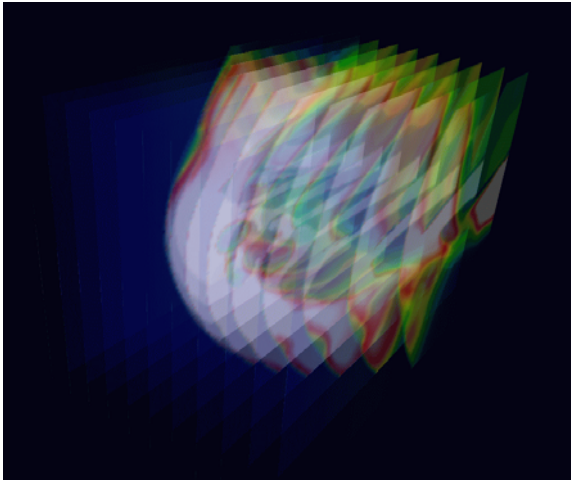


fig.3-1 : fa410zx.gif
(made from fa410zx.wrl)
Image of multiple zx planes

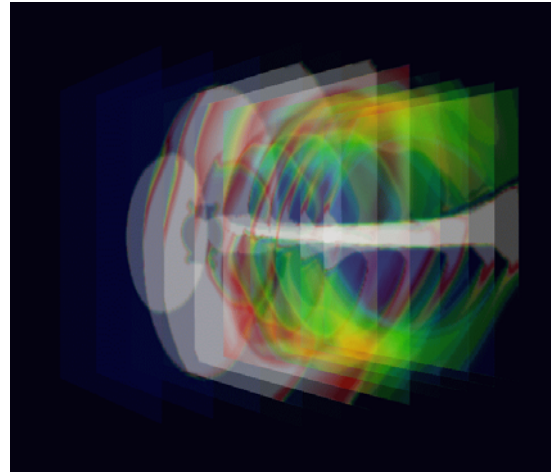


fig.3-2 : fa410yz.gif
(made from fa410yz.wrl)
Image of multiple yz planes

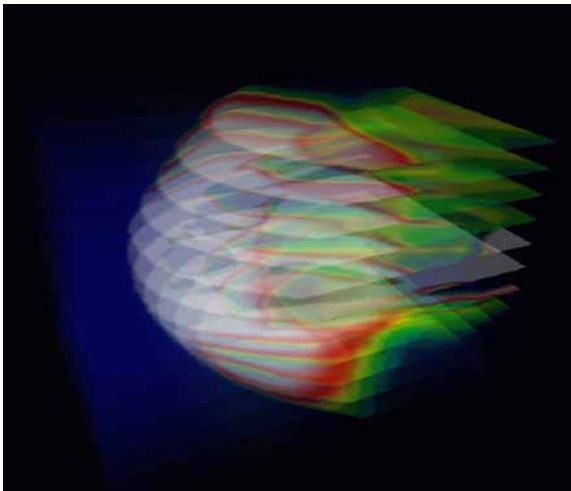


fig.3-3 : fa410xy.gif
(made from fa410xy.wrl)
Image of multiple xy planes

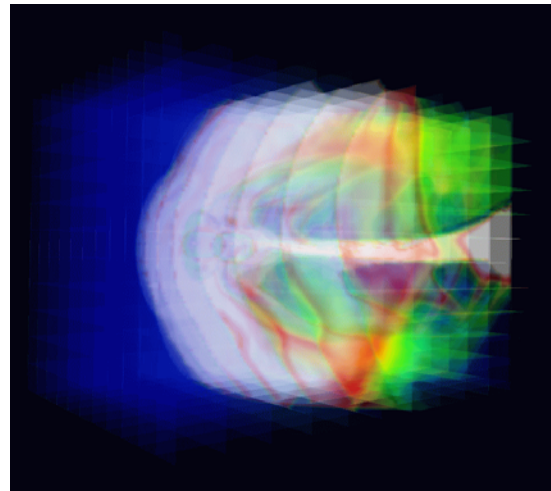


fig.3-4 : fa410xyz.gif
(made from fa410xyz.wrl)
Synthesis image with multiple planes
in three directions

4. 3-Dimensional MHD Simulation of Earth's Magnetosphere

<Example to execute the MHD Code and Graphic programs>

We will demonstrate how to execute the 3-Dimensional magnetohydrodynamic (MHD) Simulation of Earth's Magnetosphere in 1/4 volume and how to use the graphics programs to make PostScript files and VRML files in this section. In the MHD model, MHD and Maxwell's equations are solved in the solar-magnetospheric coordinate system by using modified leap-frog method when the upstream solar wind and interplanetary magnetic field (IMF) boundary conditions are given. Moreover, north-south symmetry and dawn-dusk symmetry are assumed, therefore it is enough to solve 1/4 volume as the simulation box. The main simulation Fortran program, earthb10.f is fully vectorized and can be executed on many kinds of computers. By executing the main MHD simulation code, a simulated binary file is produced as output. When the output binary file is used as input, graphics programs can be executed to make PostScript files and VRML files for three dimensional visualization.

main program : earthb10.f

earthb10.f using modified leap-frog scheme
3D MHD simulation of 1/4 earth's magnetosphere
Cartesian coordinate finite resistivity 45 degree boundary

graphics program to make PostScript files

1. gm150b.f (main) + gsub150.f (subroutine)
noon-midnight meridian and equatorial plots (black and white)
2. gm220b.f (main) + gsub220.f (subroutine)
energy distribution of cross section
3. gm480b.f (main) + gsub480.f (subroutine)
3-dimensional magnetic field lines

3-dimensional graphics program by VRML files

<Virtual Reality Modeling Language>

1. zvrmagb.f (main) + zvrsubbf (subroutine)
3-dimensional magnetic field lines
2. zvrcrib.f (main) + zvrsubb.f (subroutine)
cross sectional pattern by pixel image

<<Summary of parameters in MHD Simulation Code>>

(nx,ny,nz)=(180,60,60)	grid number without boundary
nxp=30	parameter to determine earth position
last=1024	number of time steps
iiq0=8	a unit of modified leap-frog scheme
iip0= 32	adjust upstream boundary condition
iis0= 1024	sampling step of data
thx=4.00	parameter to adjust time step
(xl,y1,zl)=(90.5,30.5,30.5)Re	length in each direction

$hx=xl/float(nx+1)=0.5Re$ grid interval in x direction
 $hy=yl/float(ny+1)=0.5Re$ grid interval in y direction
 $hz=zl/float(nz+1)=0.5Re$ grid interval in z direction
 $t=0.5*hx*thx$ time interval
 $t(real)=t*ts$ real time to one time step advance
 $=0.5*0.5*4.00*0.937$ ts is normalization value in time
 $=0.937$ sec

$x=0.5*hx*float(2*i-nx2-1+2*nxp)$ x position versus grid number
 $y=0.5*hy*float(2*j-3)$ y position versus grid number
 $z=0.5*hz*float(2*k-3)$ z position versus grid number

where $nx2=nx+2$, $ny2=ny+2$ and $nz2=nz+2$

$ro01=5.0E-4$ (5/cc) mass density of solar wind
 $pr01=3.56E-8$ pressure of solar wind
 $vsw=0.044$ (300km/s) speed of solar wind
 $bis=CP(11)=1.5E-4$ (5nT) amplitude of IMF

$eatt$ resistivity
 $rmuu$ viscosity
 $eud0$ friction or collision term

1-dimensional array variable $f(i1)=f(i,j,k,m)$

$n1=nx+2, n2=n1*(ny+2), n3=n2*(nz+2)$
 $nb=8, nbb=11, n4=n3*nb, n5=n3*nbb$

$i1=i+n1*(j-1)+n2*(k-1)+n3*(m-1)$

$m=1$: rho, plasma density
 $m=2$: V_x
 $m=3$: V_y
 $m=4$: V_z
 $m=5$: P , plasma pressure
 $m=6$: B_x
 $m=7$: B_y
 $m=8$: B_z

<<execution of main program>>

1. f77 -O earthb10.f
2. a.out &

where file must be defined in open statement like

```
c      open(10,file='earthb10.data',
c      1          access='sequential',form='unformatted')
      open(11,file='earthb11.data',
      1          access='sequential',form='unformatted')
c
```

or

1. f77 -o earthb10 -O earthb10.f
2. earthb10 &

<<execution of PostScript graphics program>>

1. f77 -c -O gsub150.f
2. f77 -O gm150b.f gsub150.o
3. a.out > gm150b.ps &
4. gs gm150b.ps
5. lp gm150b.ps

1. f77 -c -O gsub220.f
2. f77 -O gm220b.f gsub220.o
3. a.out > gm220b.ps &

1. f77 -c -O gsub480b.f
2. f77 -O gm480b.f gsub480b.o
3. a.out & : output is written in fort.10

<<execution of VRML graphics program>>

1. f77 -c -O zvrsubb.f
2. f77 -O zvrmagb.f zvrsubb.o
3. a.out & : output is written in fort.10
4. mv fort.10 fort.102

1. f77 -c -O zvrsubb.f
5. f77 -O zvrrob.f zvrsubb.o
6. a.out & : output is written in fort.10
7. mv fort.10 fort.101
8. cat fort.101 fort.102 > zvrml01.wrl

References:

- T. Ogino, A three-dimensional MHD simulation of the interaction of the solar wind with the earth's magnetosphere: The generation of field-aligned currents, *J. Geophys. Res.*, 91, 6791-6806 (1986).
- T. Ogino, R.J. Walker and M. Ashour-Abdalla, A global magnetohydrodynamic simulation of the magnetosheath and magnetopause when the interplanetary magnetic field is northward, *IEEE Transactions on Plasma Science*, Vol.20, No.6, 817-828 (1992).
- T. Ogino, Two-Dimensional MHD Code, (in *Computer Space Plasma Physics*), Ed. by H. Matsumoto and Y. Omura, Terra Scientific Publishing Company, 161-215, 411-467 (1993).
- T. Ogino, R.J. Walker and M. Ashour-Abdalla, A global magnetohydrodynamic simulation of the response of the magnetosphere to a northward turning of the interplanetary magnetic field, *J. Geophys. Res.*, Vol.99, No.A6, 11,027-11,042 (1994).

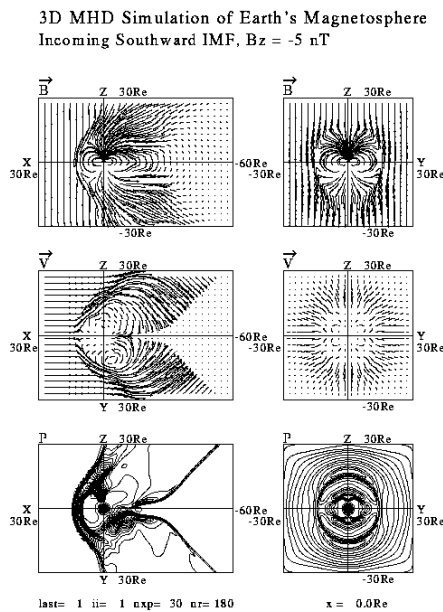


Fig.1. gm150b.gif
(converted from gm150b.ps)

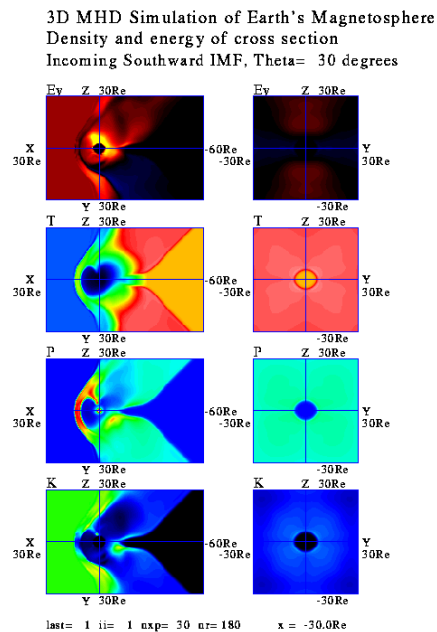


Fig.2. gm220b.gif
(converted from gm220b.ps)

3D MHD Simulation of Earth's Magnetosphere

$B_z = 0.0\text{nT}$ $N_{sw} = 5/\text{cc}$ $V_{sw} = 300\text{km/s}$ $t = 180\text{m}$

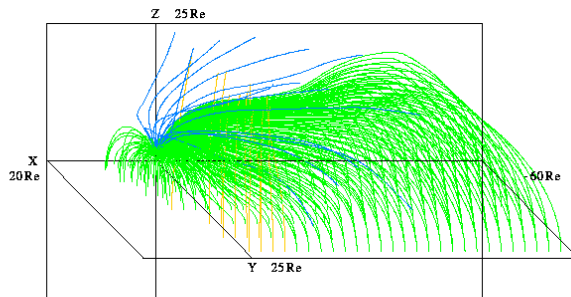


Fig.3. gm480b.gif (converted from gm480b.ps)

3D MHD Simulation of Earth's Magnetosphere

$B_z = -5.0\text{nT}$ $N_{sw} = 5/\text{cc}$ $V_{sw} = 300\text{km/s}$ $t = 64\text{m} (0\text{m})$

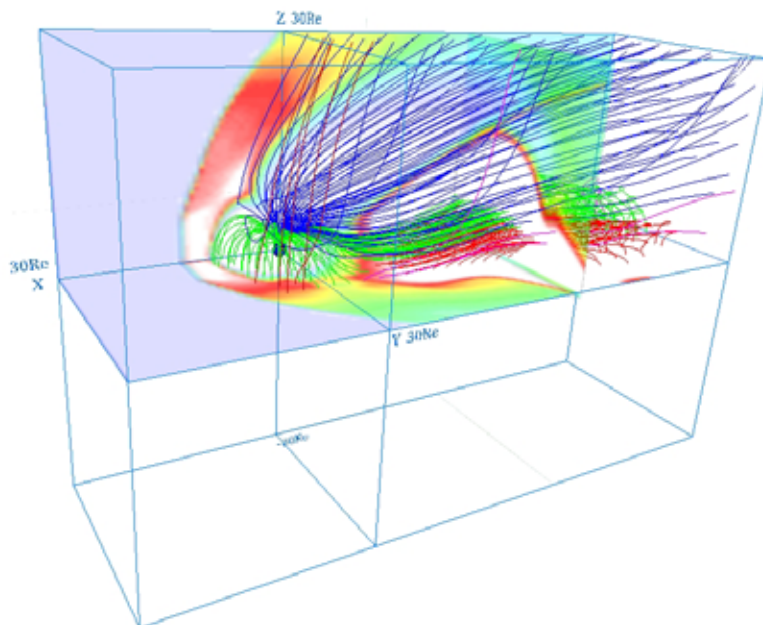


Fig.4. zvrml1b.jpg (made from zvrml1b.wrl)

5. Visualization to Various 3-Dimensional MHD Models of Earth's Magnetosphere (English)

We will show how to use the graphics programs to make VRML files of various 3-Dimensional MHD models of Earth's Magnetosphere. The output binary files produced by various MHD simulation codes by using modified leap-frog scheme are used as input.

<<execution of VRML graphics program>>

1. f77 -c -O zvrsubb.f
2. f77 -O zvrmagb.f zvrsubb.o

```
where file must be defined in open statement like
c      open(10,file='earthb10.data',
c      1      access='sequential',form='unformatted')
c      open(11,file='earthb11.data',
c      1      access='sequential',form='unformatted')
c
```

3. a.out & : output is written in fort.10
4. mv fort.10 fort.102

1. f77 -c -O zvrsubb.f
5. f77 -O zvrsubb.f zvrsubb.o
6. a.out & : output is written in fort.10
7. mv fort.10 fort.101
8. cat fort.101 fort.102 > zvrml01.wrl

<<Summary of parameters in MHD Simulation Code>>

(nx,ny,nz) = (180,60,60) grid number without boundary
nxp=30 parameter to determine earth position
(xxl,yyl,zzl) = (90.5,30.5,30.5)Re length in each direction
hx=xxl/float(nx+1)=0.5Re grid interval in x direction
hy=yyl/float(ny+1)=0.5Re grid interval in y direction
hz=zzl/float(nz+1)=0.5Re grid interval in z direction

1-dimensional array variable $f(i1)=f(i,j,k,m)$
 $i1=i+n1*(j-1)+n2*(k-1)+n3*(m-1)$

m=1 : rho, plasma density
m=2 : Vx, x component of velocity
m=3 : Vy, y component of velocity
m=4 : Vz, z component of velocity
m=5 : P, plasma pressure
m=6 : Bx, x component of magnetic field
m=7 : By, y component of magnetic field
m=8 : Bz, z component of magnetic field

5-a. Half volume model of earth's magnetosphere with IMF B_y and B_z components

<<example data file used in this section>>

gg220410.data

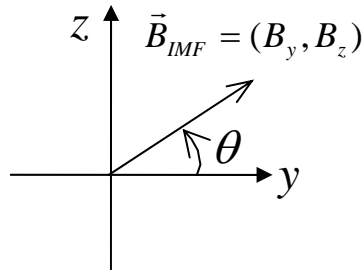
(nx,ny,nz) = (180,60,120)

nxp = 30

time = 410m

$\vec{B}_{IMF} = (B_y, B_z)$

theta = 30 deg



<<3-dimensional graphics program by VRML files>>

1. 3-dimensional magnetic field lines

zvrnaga.f (main) + zvrsuba.f (subroutine) > fa410m.wrl

2. cross sectional pattern by pixel image

zvrcoa.f (main) + zvrsuba.f (subroutine) > fa410c.wrl

3. cross sectional pattern by velocity vectors

zvrvela.f (main) + zvrsuba.f (subroutine) > fa410v.wrl

a synthesis file with above three files using "cat" > fa410w.wrl

4. 3-dimensional pixel image with multi-planes

zvrmpxa.f (main) + zvrsuba.f (subroutine)
zx planes selected in parameter > fa410zx.wrl

yz planes selected in parameter > fa410yz.wrl

xy planes selected in parameter > fa410xy.wrl

a synthesis file with above three files using "cat" > fa410xyz.wrl

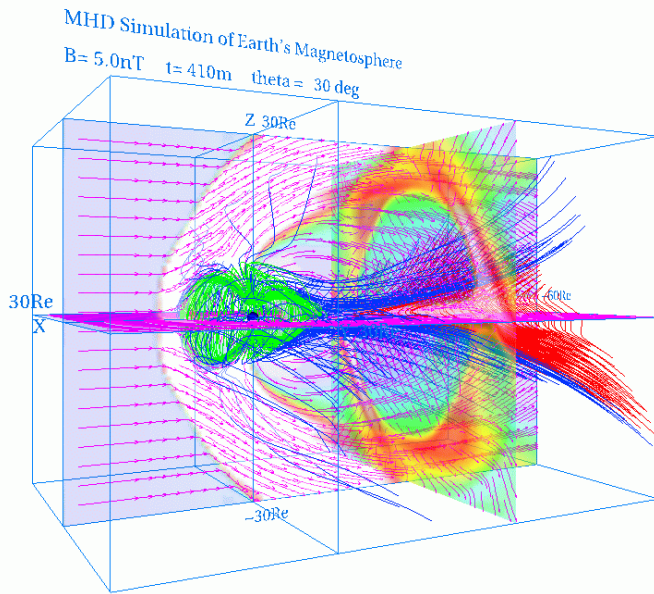


fig.5-1 : fa410w.gif (made from fa410w.wrl)
Whole image of magnetosphere

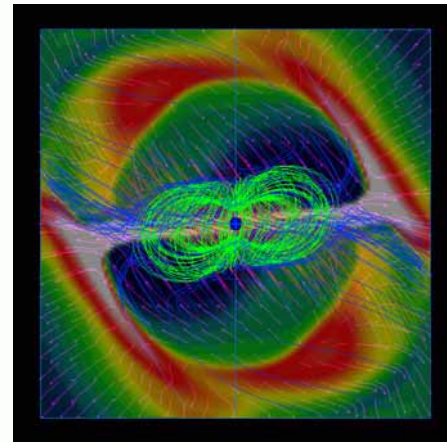


fig.5-2 : fa410s.gif
(made from fa410w.wrl)
View from the sun

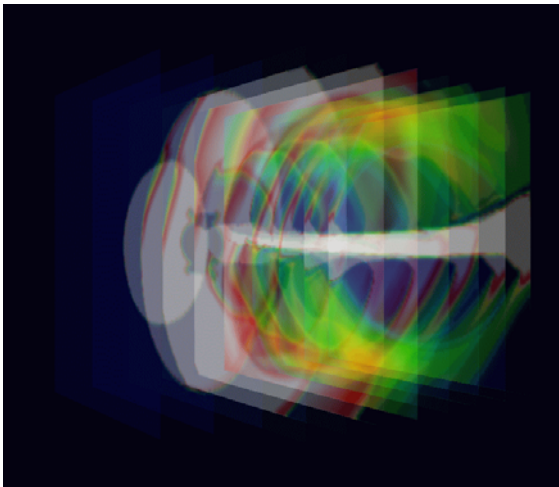


fig.5-3 : fa410yz.gif (made from fa410yz.wrl)
Image of multiple yz planes

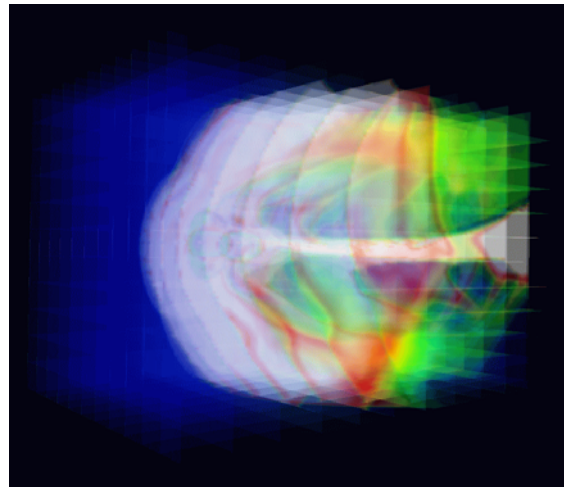


fig.5-4 : fa410xyz.gif (made from fa410xyz.wrl)
Synthesis image with multiple planes
in three directions

5-b. Quarter volume model of earth's magnetosphere

<<example data file used in this section>>

ggswb240.data

(nx,ny,nz) = (180,60,60)

nxp = 30

time = 240m

IMF B_z ($B_y = 0$)

<<3-dimensional graphics program by VRML files>>

1. 3-dimensional magnetic field lines

zvrmagb.f (main) + zvrsubb.f (subroutine) > fb240m.wrl

2. cross sectional pattern by pixel image

extended pattern to whole volume

zvrcrib.f (main) + zvrsubb.f (subroutine) > fb240c.wrl

quater volume pattern

zvrcribq.f (main) + zvrsubb.f (subroutine) > fbq240c.wrl

3. cross sectional pattern by velocity vectors

extended pattern to whole volume

zvrvelb.f (main) + zvrsubb.f (subroutine) > fb240v.wrl

quater volume pattern

zvrvelbq.f (main) + zvrsubb.f (subroutine) > fbq240v.wrl

a synthesis file with fb240m.wrl , fb240c.wrl and fb240v.wrl using "cat" > fb240w.wrl

a synthesis file with fb240m.wrl , fbq240c.wrl and fbq240v.wrl using "cat" > fbq240w.wrl

4. 3-dimensional pixel image with multi-planes

zvrmpxb.f (main) + zvrsubb.f (subroutine)

zx planes selected in parameter > fb240zx.wrl

yz planes selected in parameter > fb240yz.wrl

xy planes selected in parameter > fb240xy.wrl

a synthesis file with above three files using "cat" > fb240xyz.wrl

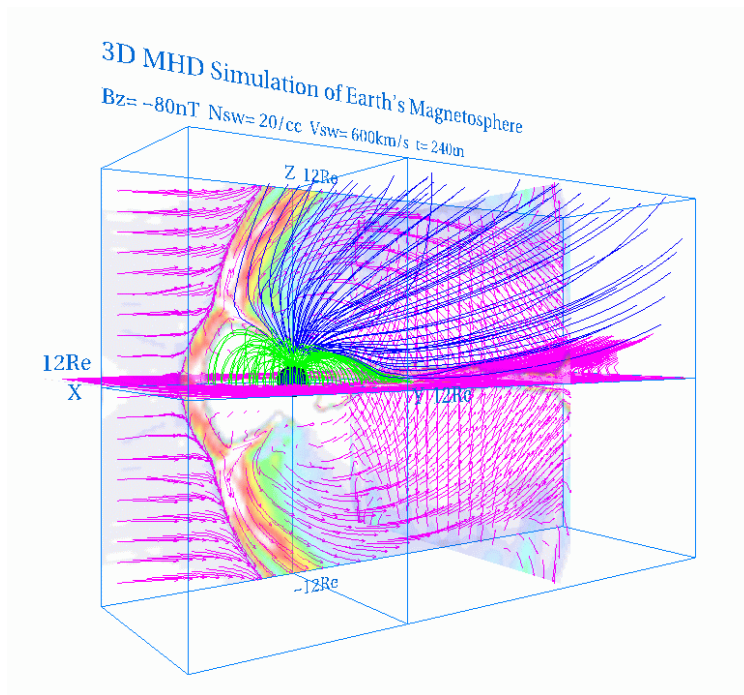


fig.5-5 : fb240w.gif (made from fb240w.wrl)
Whole image of magnetosphere

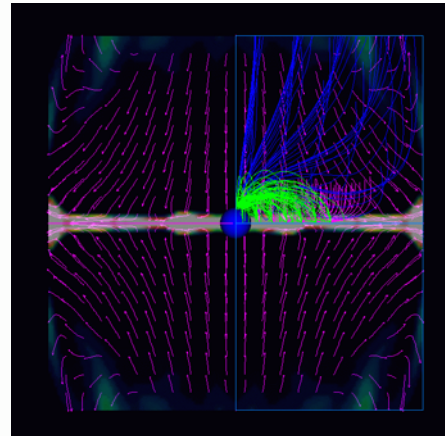


fig.5-6 : fb240s.gif
(made from fb240w.wrl)
View from the sun

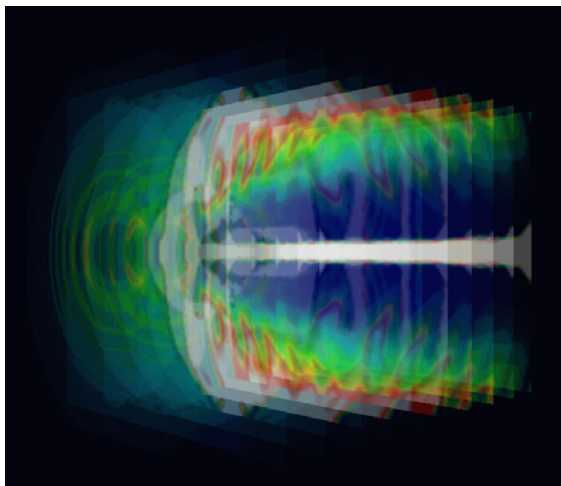


fig.5-7 : fb240yz.gif (made from fb240yz.wrl)
Image of multiple yz planes

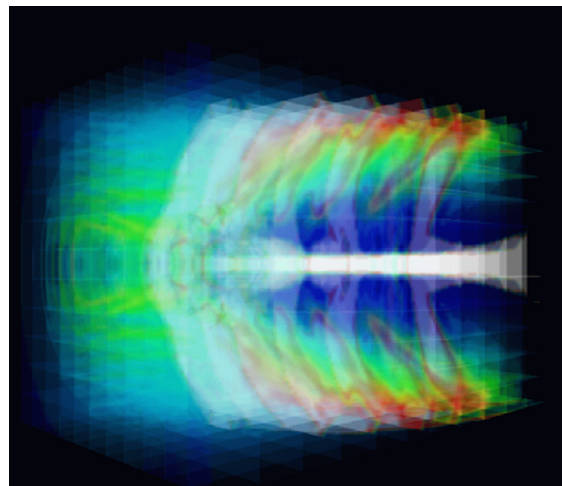


fig.5-8 : fb240xyz.gif (made from fb240xyz.wrl)
Synthesis image with multiple planes
in three directions

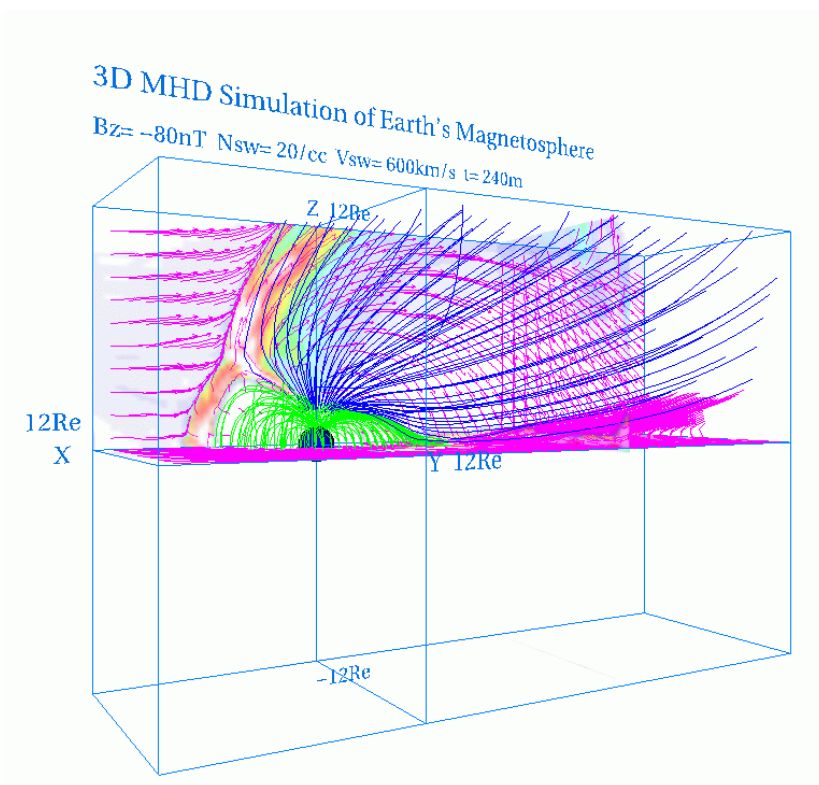


fig.5-9 : fbq240w.gif (made from fbq240w.wrl)
Whole image of magnetosphere

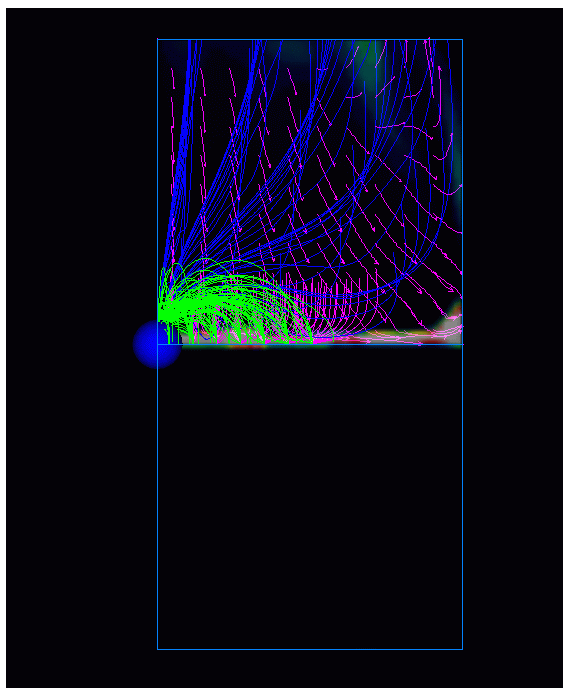


fig.5-10 : fbq240s.gif (made from fbq240w.wrl)
View from the sun

5-c. Half volume model of earth's magnetosphere with dipole tilt

<<example data file used in this section>>

gcc1a360.data

(nx,ny,nz) = (180,60,120)

nxp = 30

time = 360m

dipole tilt + IMF B_z ($B_y = 0$)

<<3-dimensional graphics program by VRML files>>

1. 3-dimensional magnetic field lines

zvrmagc.f (main) + zvrsubc.f (subroutine) > fc360m.wrl

2. cross sectional pattern by pixel image

zvrroc.f (main) + zvrsubc.f (subroutine) > fc360c.wrl

3. cross sectional pattern by velocity vectors

zvrvelc.f (main) + zvrsubc.f (subroutine) > fc360v.wrl

a synthesis file with above three files using "cat" > fc360w.wrl

4. 3-dimensional pixel image with multi-planes

zvrmpxc.f (main) + zvrsubc.f (subroutine)

zx planes selected in parameter > fc360zx.wrl

yz planes selected in parameter > fc360yz.wrl

xy planes selected in parameter > fc360xy.wrl

a synthesis file with above three files using "cat" > fc360xyz.wrl

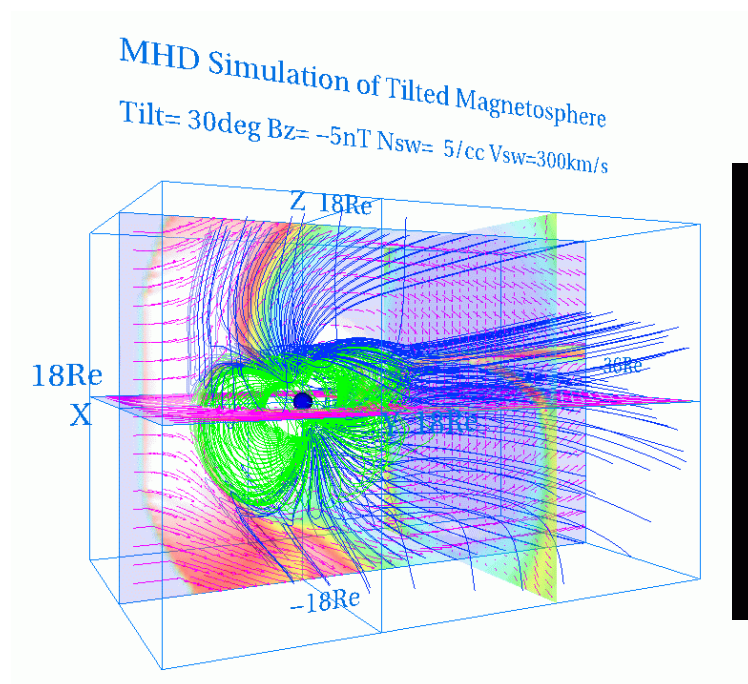


fig.5-11 : fc360w.gif (made from fc360w.wrl)
 Whole image of magnetosphere

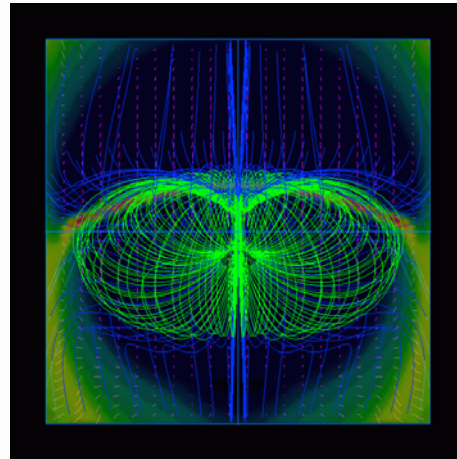


fig.5-12 : fc360s.gif
 (made from fc360w.wrl)
 View from the sun

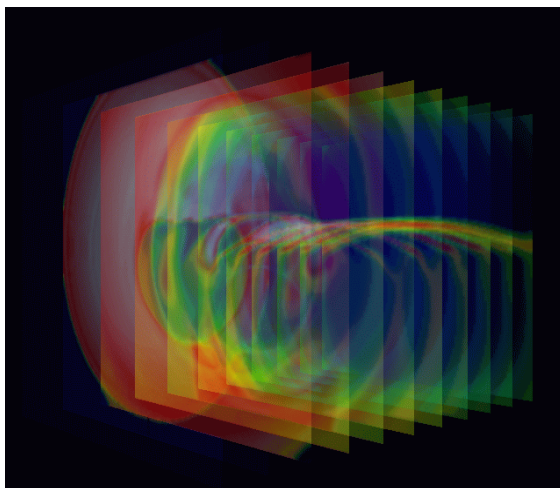


fig.5-13 : fc360yz.gif (made from fc360yz.wrl)
 Image of multiple yz planes

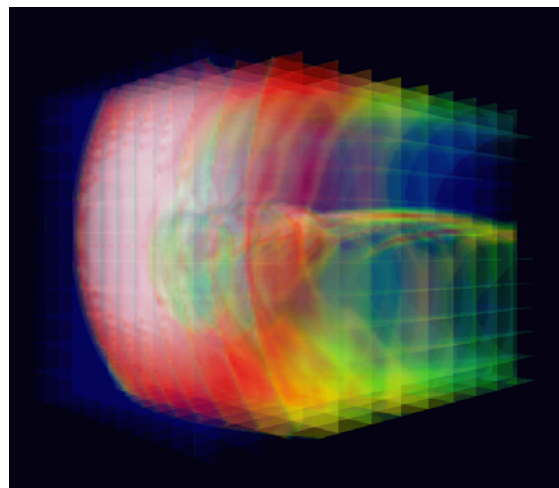


fig.5-14 : fc360xyz.gif (made from fc360xyz.wrl)
 Synthesis image with multiple planes
 in three directions

5-d. Whole volume model of earth's magnetosphere

<<example data file used in this section>>

gdd1e180.data

(nx,ny,nz) = (180,120,120)

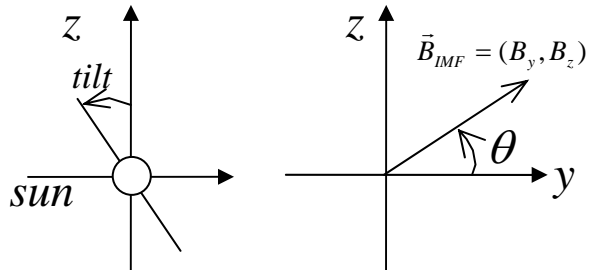
nxp = 30

time = 180m

dipole tilt + IMF $\vec{B}_{IMF} = (B_y, B_z)$

tilt = 30 deg

theta = 45deg



<<3-dimensional graphics program by VRML files>>

1. 3-dimensional magnetic field lines

zvrmag.f (main) + zvrsubd.f (subroutine) > fde180m.wrl

2. cross sectional pattern by pixel image

zvrcred.f (main) + zvrsubd.f (subroutine) > fde180c.wrl

3. cross sectional pattern by velocity vectors

zvrvel.f (main) + zvrsubd.f (subroutine) > fde180v.wrl

a synthesis file with above three files using "cat" > fde180w.wrl

4. 3-dimensional pixel image with multi-planes

zvrmpxd.f (main) + zvrsubd.f (subroutine)
zx planes selected in parameter > fde180zx.wrl

yz planes selected in parameter > fde180yz.wrl

xy planes selected in parameter > fde180xy.wrl

a synthesis file with above three files using "cat" > fde180xyz.wrl

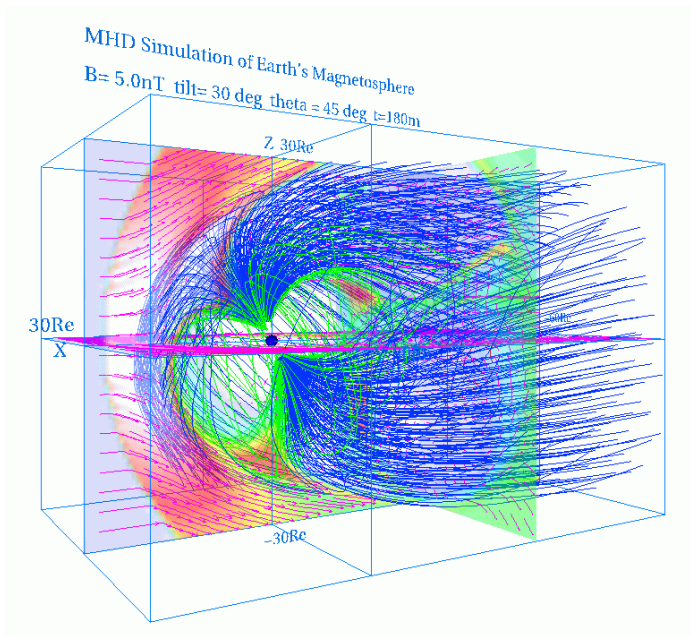


fig.5-15 : fde180w.gif (made from fde180w.wrl)
Whole image of magnetosphere

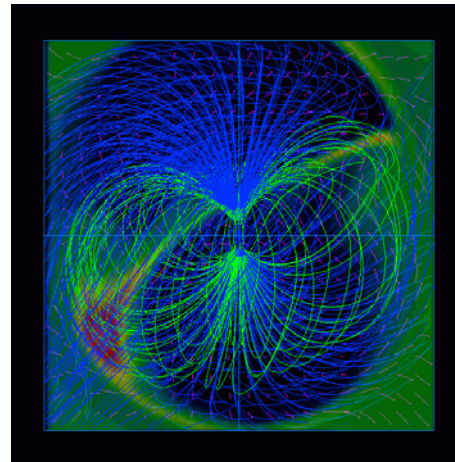


fig.5-16 : fde180s.gif
(made from fde180w.wrl)
View from the sun

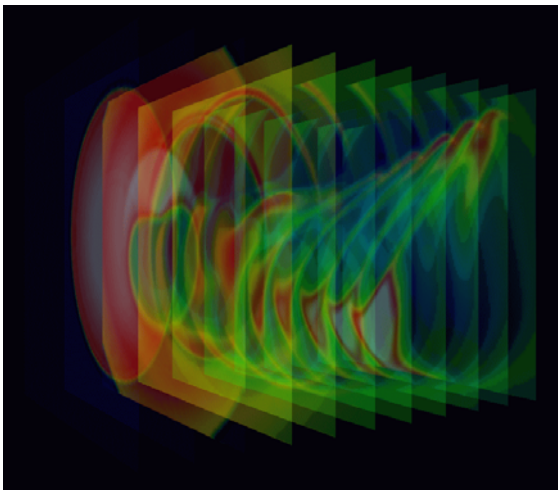


fig.5-17 : fde180yz.gif
(made from fde180yz.wrl)
Image of multiple yz planes

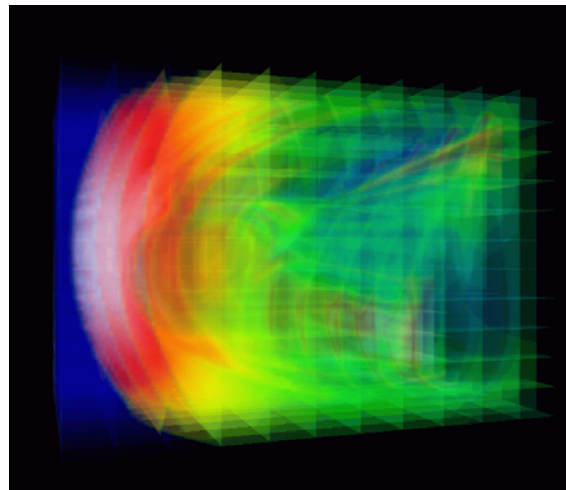


fig.5-18 : fde180xyz.gif
(made from fde180xyz.wrl)
Synthesis image with multiple planes
in three directions

6 . Application of VRML to 3-Dimensional MHD Models of Earth's Magnetosphere (English)

6-a. 3-dimensional visualization of earth's magnetosphere with multiple planes

<<example data file used in this section>>

idd1i540.data

(nx,ny,nz) = (268,134,134)

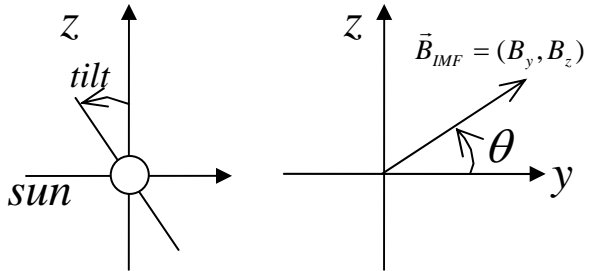
nxp = 30

time = 540m

dipole tilt + IMF $\vec{B}_{IMF} = (B_y, B_z)$

tilt = 30deg

theta = 0deg



<<3-dimensional graphics program by VRML files>>

1. 3-dimensional magnetic field lines

zvrmagd.f (main) + zvrsubd.f (subroutine) > zvrmagdc2.idd1i540.wrl

2. Plasma velocity vectors in cross section

zvrveld.f (main) + zvrsubd.f (subroutine) > zvrveldc.idd1i540.wrl

3. Pixel image of plasma temperate with multiple planes

zvrmpxd.f (main) + zvrsubd.f (subroutine)

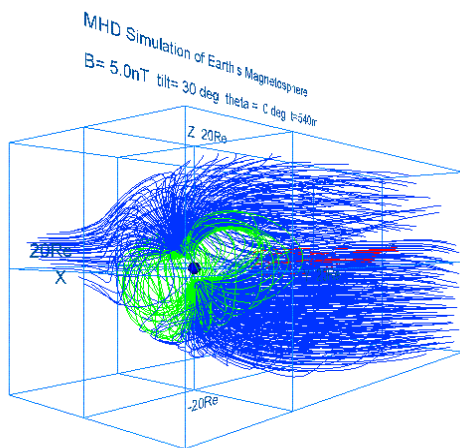
zx planes selected in parameter > pxdx. idd1i540.wrl

yz planes selected in parameter > pxdyz. idd1i540.wrl

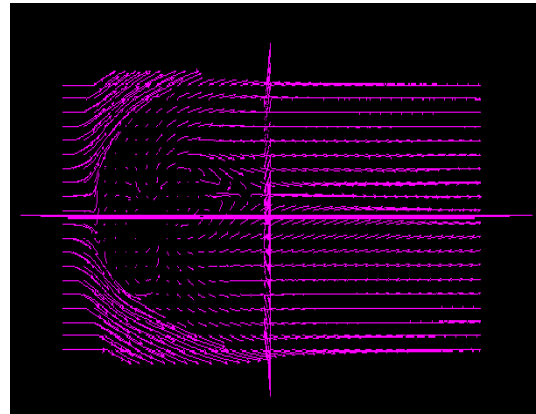
xy planes selected in parameter > pxdxy idd1i540.wrl

a synthesis file with above three files using "cat" > pxd4. idd1i540.wrl

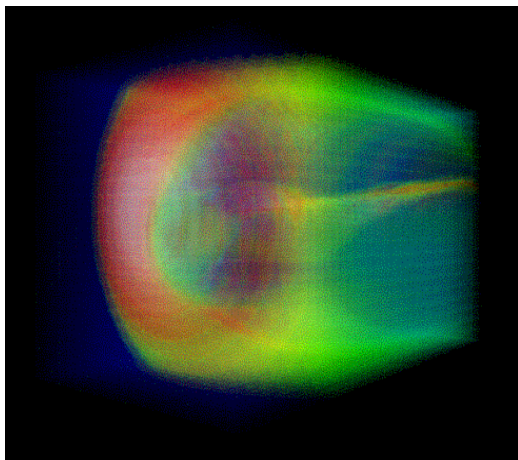
4. a synthesis file with above three files using "cat" > pxd4mdc2v.idd1i540.wrl



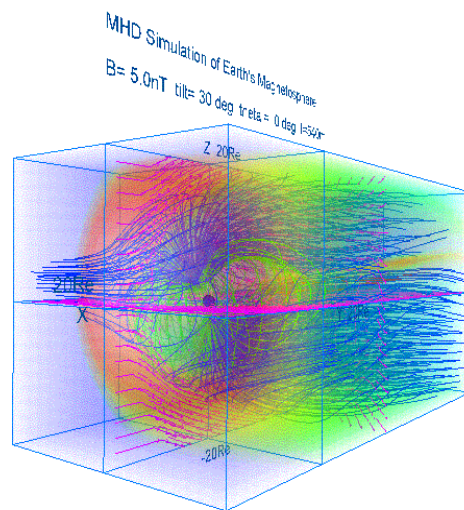
zvmagdc2.idd1i540.gif
 (made from zvmagdc2.idd1i540.wrl)
 3-dimensional magnetic field lines



zrveldc.idd1i540.gif
 (made from zrveldc.idd1i540.wrl)
 Plasma velocity vectors in cross section



pxd4.idd1i540.gif
 (made from pxd4.idd1i540.wrl)
 Pixel image of plasma temperature
 with multiple planes



pxd4mdc2v.idd1i540.gif
 (made from pxd4mdc2v.idd1i540.wrl)
 Synthesis image with multiple planes
 in three directions

6-b. 3-dimensional visualization of earth's magnetosphere with equivalence planes

<<example data file used in this section>>

idd1n720.data

(nx,ny,nz) = (268, 134, 134)

nxp = 30

time = 720m

dipole tilt + IMF

tilt = 30deg

theta = 90deg

<<3-dimensional graphics program by VRML files>>

1. 3-dimensional magnetic field lines

zvrmagd.f (main) + zvrsubd.f (subroutine) > zvrmagd. idd1n720.wrl

2. Plasma velocity vectors in cross section

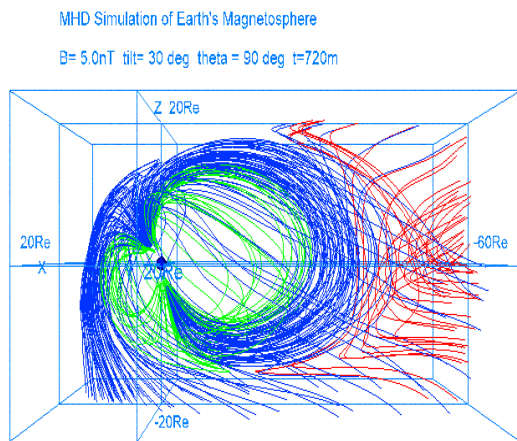
zrveld.f (main) + zvrsubd.f (subroutine) > zrveld. idd1n720.wrl

3. 3-dimensional visualization with equivalence planes

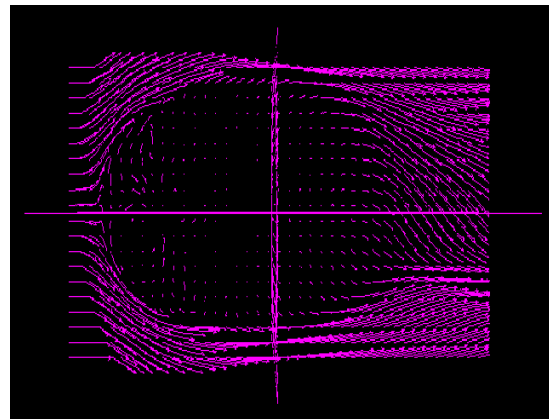
mcubvrm.f (main) + mcubsub.f (subroutine) > mcub. idd1n720.wrl

4. a synthesis file with above three files using "cat"

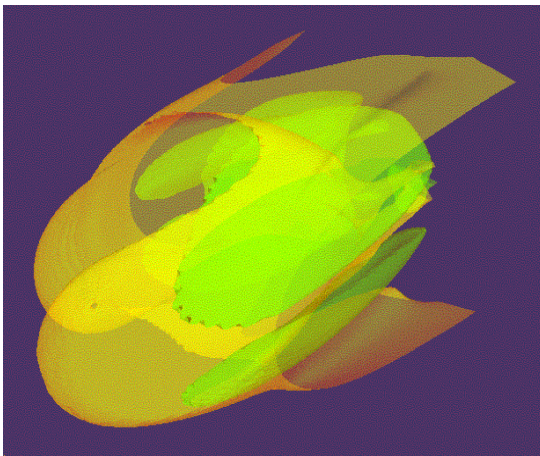
> mcubmagdv. idd1n720.wrl



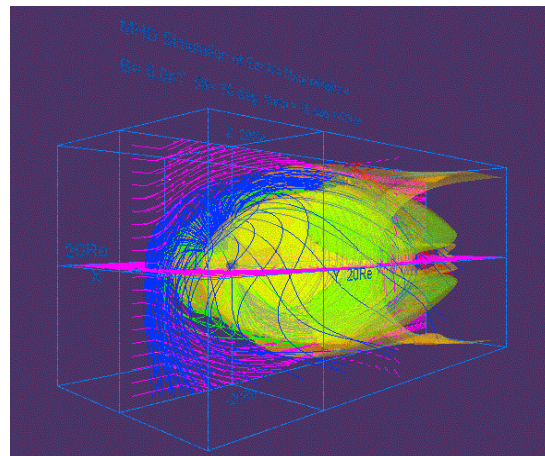
zvrmagdc2.idd1n720.gif
 (made from zvrmagdc2.idd1n720.wrl)
 3-dimensional magnetic field lines



zvrveldc.idd1n720.gif
 (made from zvrveldc.idd1n720.wrl)
 Plasma velocity vectors in cross section



mcub.idd1n720.gif
 (made from mcub.idd1n720.wrl)
 3-dimensional visualization
 with equivalence planes



mcubdc2v.idd1n720.gif
 (made from mcubmagdv.idd1n720.wrl)
 Synthesis image with equivalence planes
 in three directions

6-c. Cropping of 3-dimensional data

main program mbikxyz.f

Input variable :

nx : Data number of the x direction
ny : Data number of the y direction
nz : Data number of the z direction
x0 : The x coordinate of the origin
y0 : The y coordinate of the origin
z0 : The z coordinate of the origin
mb : The number of cropping

Description : This program produce a reduced array, fb(jx,jy,jz) from the original away, fa(ix,iy,iz), where mb is the number of cropping.

$$\begin{aligned}ix &= 1+mb*(jx-1) \\iy &= 1+mb*(jy-1) \\iz &= 1+mb*(jz-1)\end{aligned}$$

7. 3-dimensional visualization of scalar physical quantity with isosurfaces by C language

The method and examples to make the VRML2.0 file for 3-dimensional visualization of scalar physical quantity with isosurfaces by C language are presented. The VRML2.0 file for visualization is practically made by that the mcube001.c is compiled and executed.

Method to use:

1. cc mcube001.c

Main program (mcube001.c etc.) is compiled to make execution file, a.out.

2. a.out output file: mcube001.wrl

Execute the execution file, a.out and make a VRML2.0 file such as mcube001.wrl etc. by putting the output file name like mcube001.wrl etc. following "output file: ".

(1) Production of Isosurfaces by Marching-Cubes Method

main program: mcube001.c

subroutine: mcube000.h

Explanation: Program to draw the isosurfaces by getting 3-dimensional scalar physical data, $v[nX][nY][nZ]$

Input parameters:

header file: mcube000.h

NX: number of grid in x-direction

NY: number of grid in y-direction

NZ: number of grid in z-direction

nX =NX+1: number of data in x-direction

nY =NY+1: number of data in y-direction

nZ =NZ+1: number of data in z-direction

main program: mcube001.c

Viewpoint(fp): function to determine view point

fp: output file name

dr_grid(fp, gc): function to draw the grid

fp: output file name

gc[3]: color of line

dr_scale(fp, gc): function to draw the scale

fp: output file name

gc[3]: color of line

mk_equ_face001(fp, th, v, c, tr): function to draw an isosurface

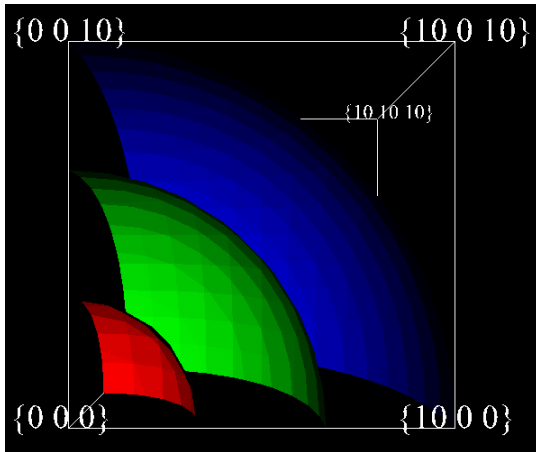
fp: output file name

th: level of isosurface

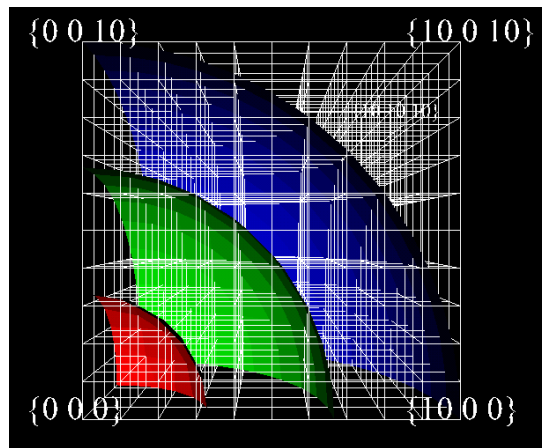
c: color of isosurface

v: 3-dimensional input data

tr: degree of transparency



Example 1: mcube0011.wrl (without grid)



Example 2: mcube0012.wrl (with grid)

(2) Production of Isosurfaces by Tetrahedrons

main program: mcube002.c

subroutine: mcube000.h

Explanation: Program to draw the isosurfaces by getting 3-dimensional scalar physical data, $v[nX][nY][nZ]$

Input parameters:

header file: mcube000.h

NX: number of grid in x-direction

NY: number of grid in y-direction

NZ: number of grid in z-direction

$nX = NX + 1$: number of data in x-direction

$nY = NY + 1$: number of data in y-direction

$nZ = NZ + 1$: number of data in z-direction

main program: mcube002.c

Viewpoint(fp): function to determine view point

fp: output file name

dr_grid(fp, gc): function to draw the grid

fp: output file name

gc[3]: color of line

mk_equ_face002(fp, th, v, c, tr): function to draw an isosurface

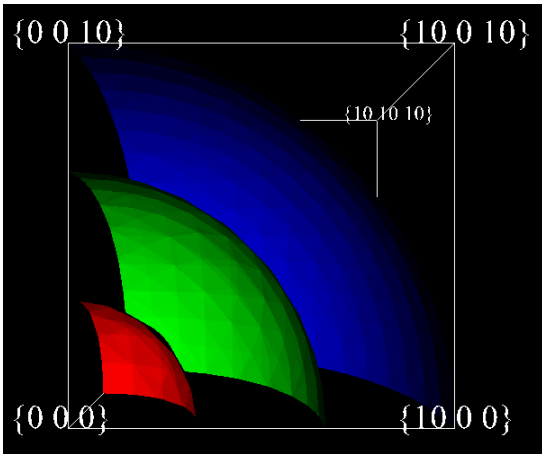
fp: output file name

th: level of isosurface

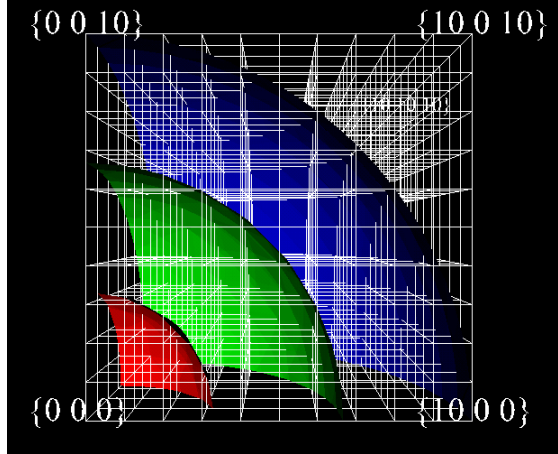
c: color of isosurface

v: 3-dimensional input data

tr: degree of transparency



Example3 : mcube0021.wrl(without grid)



Example4 : mcube0022.wrl(with grid)

References

VRML2.0 3D Cyber Space Language, by Kenjiro Miura, Asakura-syoten, 1996

Web3D Graphics, by Tetsuo Hirouchi, Piason Education, 2001

The VRML 2.0 Handbook, by Jed Hartman and Josie Wernecke, Silicon Graphic, Inc,
Addison-Wesley Publishing Co., 1996

The Annotated VRML 2.0 Reference Manual, by Rikk Carey and Gavin Bell,
Addison-Wesley Publishers Japan, 1998

PostScript Reference Manual Second Edition, by Adobe Systems, ASCII Corporation, 1995

PostScript by Example, by H. McGilton and M. Campione,
Addison-Wesley Publishing Co., 1992

Fortran program and image file list

1. Fundamental example (1)

No.	Contents	Fortran Program	VRML file	gif file
	Subroutine package	pk1subrtn.f		
1	Cube	box1.f	box1.wrl	box1.gif
2	Rectangular parallelepiped	box2.f	box2.wrl	box2.gif
3	Trigonal pyramid	cone1.f	cone1.wrl	cone1.gif
4	Column	cylinder1.f	cylinder1.wrl	cylinder1.gif
5	Sphere	sphere1.f	sphere1.wrl	sphere1.gif
6	Background color	backgrd4.f	backgrd4.wrl	backgrd4.gif
7	Character string	text1.f	text1.wrl	text1.gif
8	Polygonal line : monochrome	lineset1.f	lineset1.wrl	lineset1.gif
9	Polygonal line : gradation	lineset2.f	lineset2.wrl	lineset2.gif
10	Dotted line : monochrome	pointset1.f	pointset1.wrl	pointset1.gif
11	Dotted line : gradation	pointset2.f	pointset2.wrl	pointset2.gif
12	Arrow with the thickness	arrow2.f	arrow2.wrl	arrow2.gif
13	Curve with the thickness	linemesh1.f	linemesh1.wrl	linemesh1.gif
14	Plane (1) : Monochromatic triangular mesh	mesh1.f	mesh1.wrl	mesh1.gif
15	Plane (2) : Triangular belt	triangb1.f	triangb1.wrl	triangb1.gif
16	Plane (3) : Polychrome triangular mesh	triangm1.f	triangm1.wrl	triangm1.gif
17	Solid : It is composed of the triangular mesh	defusem1.f	defusem1.wrl	defusem1.gif
18	Texture mapping	image1.f	image1.wrl	image1.gif
19	The complicated curved surface : Triangular mesh	ebmesh3d.f	ebmesh3d.wrl	ebmesh3d.gif

2. Fundamental example (2)

No.	Contents	Fortran Program	VRML file	gif file	jpg file	rgb file
	Subroutine package	zvrsuba.f				
1	Character	msymbol.f	msymbol.wrl	msymbol.gif		
2	Point	mpoint.f	mpoint.wrl	mpoint.gif		
3	Line	mline.f	mline.wrl	mline.gif		
4	Triangular mesh	mtriangm.f	mtriangm.wrl	mtriangm.gif	mtriangm.jpg	mtriangm.rgb
5	Pixel image	mzpt03.f	mzpt03.wrl	mzpt03.gif	mzpt03.jpg	mzpt03.rgb
6	Frame and pixel image and character	mpix015.f	mpix015.wrl	mpix015.gif	mpix015.jpg	mpix015.rgb
7	Equivalence plane	mcube301.f	mcube301.wrl	mcube301.gif	mcube301.jpg	mcube301.rgb
8	Outer magnetosphere by 3-dimensional lattice	outline3d.f	outline3d.wrl	outline3d.gif	outline3d.jpg	
9	Inner magnetosphere by 3-dimensional lattice	inline3d.f	inline3d.wrl	inline3d.gif	inline3d.jpg	
10	Ionosphere by 3-dimensional lattice	denline3d.f	denline3d.wrl	denline3d.gif	denline3d.jpg	

3. Application to the MHD simulation

No.	Contents	Fortran Program	VRML file	gif file
	Subroutine package	zvrsuba.f		
1	Sectional views such as plasma temperature of the earth's magnetosphere are drew by the pixel image <ul style="list-style-type: none"> The pixel image using maximum value and minimum value 	zvrCroa.f	zvrCroa.wrl	zvrCroa.gif
2	Three-dimensional structure of the earth's magnetosphere <ul style="list-style-type: none"> The decision by beforehand describing line of magnetic force Three-dimensional line of magnetic force starting from polar region Three-dimensional line of magnetic force 	zvrMaga.f	zvrMaga.wrl	zvrMaga.gif

	starting from equatorial plane			
	<ul style="list-style-type: none"> The test of the boundary in the closed region with that it opens 			
3	The synthesis of the three-dimensional image of pixel image and line of magnetic force		zvr01.wrl	zvr01.gif
	Subroutine package	zvrsuba.f		
4	Three dimensional distribution of the plasma temperature in the whole earth's magnetosphere drawn by the multiple pixel image planes	zvrmpxa.f	fa410zx.wrl fa410yz.wrl fa410xy.wrl fa410xyz.wrl	fa410zx.gif fa410yz.gif fa410xy.gif fa410xyz.gif

4.3-Dimensional MHD Simulation of Earth's Magnetosphere

(Example to execute the MHD Code and Graphic programs)

No.	Contents	Fortran Main Program	Subroutine Package	PostScript file	VRML file or gif file
	3D MHD simulation code of 1/4 earth's magnetosphere	earthb10.f			
	Graphics program to make PostScript files				
1	noon-midnight meridian and equatorial plots (black and white)	gm150b.f	gsub150.f	gm150b.ps	gm150b.gif
2	energy distribution of cross section	gm220b.f	gsub220.f	gm220b.ps	gm220b.gif
3	3-dimensional magnetic field lines	gm480b.f	gsub480.f	gsub480.ps	gsub480.gif
	3-dimensional graphics program by VRML files (Virtual Reality Modeling Language)				
1	3-dimensional magnetic field lines	zvrmagb.f	zvrsubb.f		zvrml1b.wrl
2	cross sectional pattern by pixel image	zvrrob.f	zvrsubb.f		zvrml1b.jpg

5. Visualization to Various 3-dimensional MHD Models of Earth's Magnetosphere

5-a. Half volume model of earth's magnetosphere with IMF By and Bz components

No.	Contents	Fortran Main Program	VRML file	gif file
	Subroutine Package	zvrsuba.f		
1	3-dimensional magnetic field lines	zvrмага.f	fa410w.wrl	fa410w.gif fa410s.gif
2	cross sectional pattern by pixel image	zvrroa.f		
3	cross sectional pattern by velocity vectors	zvrvela.f		
4	3-dimensional pixel image with multi-planes	zvrmpxa.f	fa410xyz.wrl fa410yz.wrl	fa410xyz.gif fa410yz.gif

5-b. Quarter volume model of earth's magnetosphere

No.	Contents	Fortran Main Program	VRML file	gif file
	Subroutine Package	zvrsubb.f		
1	3-dimensional magnetic field lines	zvrmagb.f	fb240w.wrl fbq240w.wrl	fb240w.gif fb240s.gif fbq240w.gif fbq240s.gif
2	cross sectional pattern by pixel image	zvrrob.f zvrrobq.f		
3	cross sectional pattern by velocity vectors	zvrvelb.f zvrvelbq.f		
4	3-dimensional pixel image with multi-planes	zvrmpxb.f		

5-c. Half volume model of earth's magnetosphere with dipole tilt

No.	Contents	Fortran Main Program	VRML file	gif file
	Subroutine Package	zvrsubc.f		

1	3-dimensional magnetic field lines	zvrmagc.f		
2	cross sectional pattern by pixel image	zvrroc.f	fc360w.wrl	fc360w.gif
3	cross sectional pattern by velocity vectors	zvrvelc.f		fc360s.gif
4	3-dimensional pixel image with multi-planes	zvrmpxc.f	fc360xyz.wrl fc360yz.wrl	fc360xyz.gif fc360yz.gif

5-d. Whole volume model of earth's magnetosphere

No.	Contents	Fortran Main Program	VRML file	gif file
	Subroutine Package	zvrsubd.f		
1	3-dimensional magnetic field lines	zvrmagd.f	fde180w.wrl	fde180w.gif
2	cross sectional pattern by pixel image	zvrrod.f		fde180s.gif
3	cross sectional pattern by velocity vectors	zrveld.f		
4	3-dimensional pixel image with multi-planes	zvrmpxd.f	fde180xyz.wrl fde180yz.wrl	fde180xyz.gif fde180yz.gif
5	3D pixel image for dipole tilt	zvrmpxde.f zvrmagde.f	icca810.wrl icca820.wrl	icca810.gif icca820.gif
6	3D pixel image for dipole tilt and IMF By	zvrmpxde.f zvrmagde.f	iddh225.wrl iddh230.wrl	iddh225.gif iddh230.gif
7	3D pixel image for IMF Bx, By and Bz	zvrmpxde.f zvrmagde.f zrvelde.f	ideca20.wrl ideda20.wrl idea20b.wrl	ideca20.gif ideda20.gif idea20b.gif

6. Application of VRML to 3-Dimensional MHD Models of Earth's Magnetosphere

6-a. 3-dimensional visualization of earth's magnetosphere with multiple planes

No.	Contents	Fortran Main Program	VRML file	gif file
	Subroutine Package	zvrsubd.f		
1	3-dimensional magnetic field lines	zvrmagd.f	zvrmagdc2.idd1i540.wrl	zvrmagdc2.idd1i540.gif
2	Plasma velocity vectors in cross section	zrveld.f	zrveldc.idd1i540.wrl	zrveldc.idd1i540.gif
3	Pixel image of plasma temperate with multiple planes	zvrmpxdc5.f	pxd4.idd1i540.wrl	pxd4.idd1i540.gif
4	Synthesis image with multiple planes in three directions		pxd4mdc2v.idd1i540.wrl	pxd4mdc2v.idd1i540.gif

6-b. 3-dimensional visualization of earth's magnetosphere with equivalence planes

No.	Contents	Fortran Main Program	Subroutine Program	VRML file
1	3-dimensional magnetic field lines	zvrmagd.f	zvrsubd.f	zvrmagdc2.idd1n720.gif
2	Plasma velocity vectors in cross section	zrveld.f	zvrsubd.f	zrveldc.idd1n720.gif
3	3-dimensional visualization with equivalence planes	mcubvrm.f	mcubsub.f	mcub.idd1n720.gif
4	Synthesis image with multiple planes in three directions			mcubdc2v.idd1n720.gif

6-c. Cropping of 3-dimensional data

No.	Contents	Fortran Main Program	Subroutine Program	VRML file
1	cropping of 3-dimensional data	mbikxyz.f		

7. 3-dimensional visualization of scalar physical quantity with isosurfaces by C language

No.	Contents	C MainProgram	Subroutine Program	VRML file	Gif file
1	Production of Isosurfaces by Marching-Cubes Method	mcube001.c	mcube000.h	mcube001.wrl	mcube0011.GIF
2	Production of Isosurfaces by Tetrahedrons	mcube002.c	mcube000.h	mcube002.wrl	mcube0021.GIF
3	Production of Isosurfaces by Marching-Cubes Method (Fortran)	mcube001.f	mcsb001.f	mcube03.wrl mcube04.wrl	mcube03.gif mcube04.gif

8. Three-dimensional visualization of data calculated from simulation

No.	Contents	Fortran MainProgram	Subroutine Program	Setting file	VRML file	GIF file
1	Making temporary file	main.f90	subs.f90	constants.f90		
2	Making VRML file	zvrmpr.f	zvrsb.f		eb1.wrl eb.wrl	eb1.gif eb.gif

Acknowledgements:

Individual subroutines in this manual, "How to Use VRML" have been developed by the project team of "Development and standardization of the visualization tool by the VRML" of the Solar-Terrestrial Environment Laboratory, Nagoya University from 1997, and it was arranged by the JST Computer Science and Technology Promotion Program, "The Development of Space Simulation Net-Laboratory System" as the original manual. Then it has been successively developed by Grants-in-Aid from the Japan Society for Promotion Science (JSPS) (A)(1) "Study of Solar Wind-Magnetosphere-Ionosphere Dynamics Using Common Parallel Computation Codes in MHD and Particle Simulation" and was finally arranged by Grant-in Aid for Publication of Scientific Research Results, "Space Weather International Collaborative Research Database in Japan" as the present manual. The computer simulation was performed on Fujitsu VPP5000-16PE in the Information Technology Center of Nagoya University

Project team of "Development and standardization of the visualization tool by the VRML"

Tatsuki Ogino (Leader)

Maki Nakao

Yuka Kadowaki

Minako Suganuma

Manami Inayoshi

Koichi Ota

Keiichiro Fukazawa

Dai Owaki

Kazuhide Sawada

March, 2004

Solar-Terrestrial Environment Laboratory, Nagoya University

3-13 Honohara, Toyokawa, Aichi 442-8507, Japan

TEL:+81-533-89-5207 FAX:+81-533-89-5090

E-mail:ogino@stelab.nagoya-u.ac.jp

