Conf-940424--17

LA-UR-94-157

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36

TITLE: GEOMETRY CREATION FOR MCNP BY SABRINA AND XSM

AUTHOR(S): K. A. Van Riper

SUBMITTED TO: Eighth International Conference on Radiation Shielding, April 24-27, 1994, Arlington, TX

Los Alamos

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

GEOMETRY CREATION FOR MCNP BY SABRINA AND XSM

Kenneth A. Van Riper
Radiation Transport Group
X-6, MS B226
Los Alamos National Laboratory
Los Alamos, NM 87545
(505)667-8104, kvr@lanl.gov

ABSTRACT

The Monte Carlo N-Particle transport code MCNP is based on a surface description of 3-dimensional geometry. Cells are defined in terms of boolean operations on signed quadratic surfaces. MCNP geometry is entered as a card image file containing coefficients of the surface equations and a list of surfaces and operators describing cells. Several programs are available to assist in creation of the geometry specification, among them Sabrina and the new "Smart Editor" code XSM. We briefly describe geometry creation in Sabrina and then discuss XSM in detail. XSM is under development; our discussion is based on the state of XSM as of January 1, 1994.

I. SABRINA

A. Overview

Sabrina is a three-dimensional geometry modeling and visualization program. Geometries may be defined using MCNP surface descriptions—a typical use is to visualize the model described by an existing MCNP input file—or by combinatorial solid geometry, where models are defined in terms of boolean operations on finite, closed bodies. Among the supported bodies are spheres, ellipsoids, wedges, cones, and 8-sided polyhedra. Tabulated bodies of rotation and extrusion are also supported. Sabrina can convert a body geometry model into a surface description suitable for input to MCNP.

An image of a geometry model may be made with proper three-dimensional perspective from any viewpoint. There are two rendering methods—ray tracing and line drawing. Line drawings display the edges and intersections of bodies; hidden lines may be removed. Two line drawings, with and without hidden line removal, are shown in Figure 1. The ray tracing mode yields color shaded images or an "etch" image where

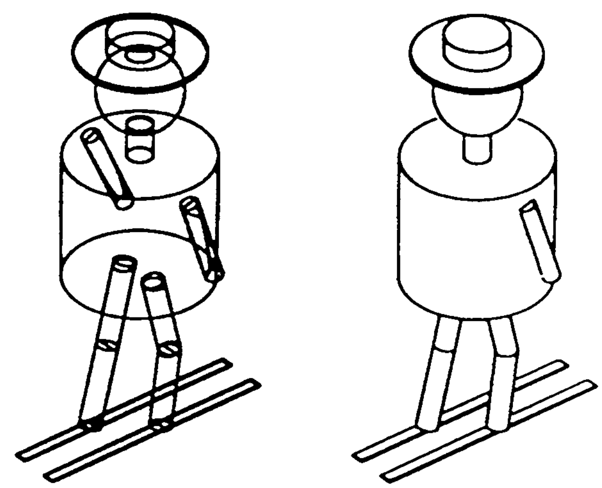

Fig. 1—A Sabrina line drawing.

lines correspond to changes in surfaces. Cutter bodies may be used in the ray-tracing methods to display the interior of a model. An "etch" ray-traced image, with a cutter body exposing the interior, is shown in Figure 2. The color and transparency of cells may be set by cell number, material, or density. Checking for geometry errors—underdefined or overdefined space—may be enabled while rendering with either method. When an error is encountered, Sabrina reports on the offending surfaces and cells and gives a visual indication of the location of the error. Sabrina can display the tracks of particles as calculated by MCNP.

A help facility supplies a few words to a few paragraphs on each command. The current version of Sabrina will execute on most UNIX machines, from simple workstations to UNICOS on CRAY computers. The

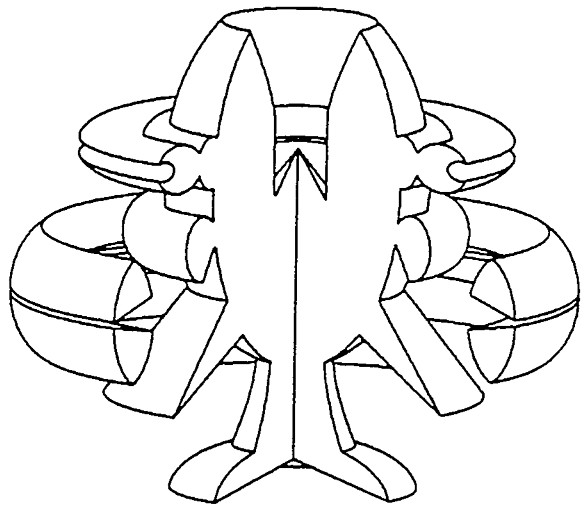

Fig. 2—A Sabrina ray-traced image.

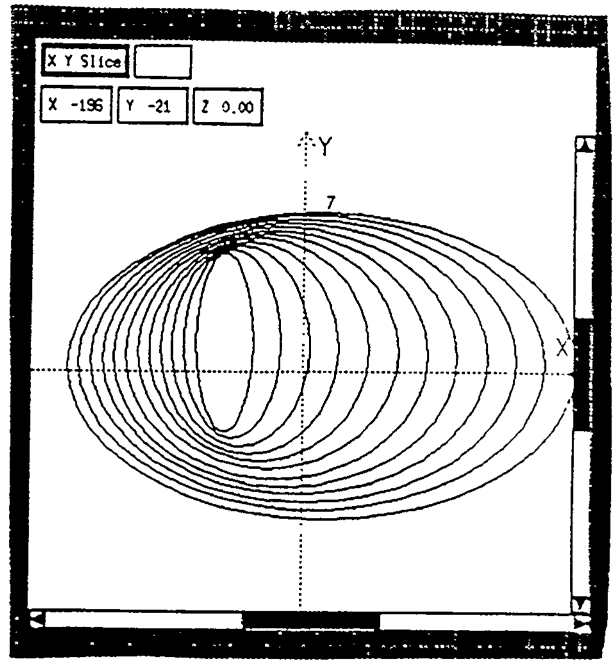

Fig. 3—The XY Slice.

X Window system is used for displaying graphics on a workstation or terminal screen. Sabrina generates PostScript files for hardcopy.

The user communicates with Sabrina through keyword-driven commands. These commands may also by read from files, and all commands input may be saved to a journal file. A graphical user interface, XSAB, is under development. XSAB is based on the OSF/MOTIF X Window toolkit. It is a separate program which generates commands for Sabrina in response to user actions—mouse motion, button presses, and keyboard input—and intercepts output from Sabrina. The communication with Sabrina is through UNIX pipes.

B. Geometry Creation in Sabrina

Sabrina was the first attempt to provide a more convenient and interactive method for creating MCNP geometry input with visual feedback. The original version of Sabrina was based solely on solid body geometry, with cells being defined in terms of boolean operations on the bodies. In many cases, the use of a body description is more intuitive than specifying the equivalent surfaces. The bodies could be displayed with the line drawing rendering (ray tracing was added later). Line drawings can be made as more bodies or cell descriptions are typed in, and bodies and cells can be modified or deleted. While certainly more interactive than preparing a complete geometry description file for batch processing, all input is with typed commands and there is no graphical manipulation of the geometry

The Sabrina user may also enter a surface geometry mode where the cells are specified in terms of surfaces This mode is equivalent to creating an MCNP input file. and has no great advantage.

II. XSM

The idea of a "Smart Editor" for creating MCNP input originated with Lee Carter. Our implementation in XSM is designed to be interactive, graphically based. and usable with a minimum of keyboard input. It uses a workstation mouse to position geometrical primative objects and define cells. XSM displays three orthogonal geometry planes, or "slices". Surfaces are shown in each slice as a line (or lines) of intersection of the surface with the slice. The *XY* slice is shown in figure 3 By selecting appropriate actions from a menu, a user may create a new surface or solid body, modify or copy an existing object, and create cells bounded by selected surfaces. Attributes and information may also be entered on cell, surface, and body property forms. Once created, a geometry may be written out in MCNP or Sabrina input format.

A. Alignment of Geometries

Most geometry objects in XSM are aligned with one of the principal coordinate axes—*X*, *Y*, and *Z* Cylinders, for example, must be parallel to an axis, and at least 1 offset vector of a *BOX* body must be so parallel. This practice follows MCNP, where simple objects are similarly aligned. MCNP provides a general quadratic surface for direct entry of arbitrarily aligned surfaces; an oblique orientation may also be achieved by use of a surface (*TR*) or cell (*TRCL*) transformation.

During development of XSM, the simplest MCNP surfaces were implemented first, both because they are the most useful and because their straightforward computation was suitable for rapid prototyping and development. In the majority of MCNP geometries we have seen, the objects are aligned with respect to the coordinate axes. In most other cases, where possible, a subassembly is built from aligned surfaces and then rotated with a transformation. Future versions of XSM may handle surfaces and bodies with arbitrary orientations.

B. Working in 2 Dimensions

Surfaces are created or modified in XSM by positioning the intersection of the surface with a plane, or slice, (bodies are created by positioning a set of surfaces). The user must have some idea of the 3-dimensional object being modeled. A full 3-dimensional view is not immediately available, but may be rendered by passing the geometry to Sabrina or to the polygonal viewer from the Frankenstein project. Either of these viewers requires some batch processing before an image is displayed. The emphasis in the XSM 2-D slices is interactive manipulation of the geometry. In either of the 3-dimensional renderings, outer objects must be made invisible in order to see the inner workings (one can see through objects in the wireframe mode of the Frankenstein polygonal renderer; this may or may not be sufficient). In XSM, the depth of a slice can be varied. The depth of one slice can be interactively varied by dragging its intersection in another slice; the image changes immediately to reflect the new slice through the model.

XSM is designed to permit graphical definition and positioning of objects. The ambiguities of graphically defining a position in 3-dimensional space on a 2-D screen are evident. Some constraint must be supplied to augment the 2-D location. In XSM, this constraint is provided by the restriction to a 2-dimensional slice of the geometry.

C. Surfaces

1. Surface Creation. To create a surface, the *Surfaces* menu item brings up a list of surface classes; a selection shows a submenu. The *PLANE* class, for instance, has a submenu consisting of planes perpendicular or parallel to the *X*, *Y*, and *Z* axes, and a plane of arbitrary orientation. Once a specific surface has been selected, an icon on each slice, the *Action Label*, shows in which slice(s) the surface may be positioned. For example, a plane perpendicular to the *X* axis may be positioned in the *XY* or *XZ* slices, but the *YZ* slice cannot be used. The user initiates the surface by pressing a mouse button and dragging the surface to the desired position. The initial mouse press will help determine the origin for non-planar surfaces. The motion is shown in each of the slices which intersect the surface. When using the left mouse button, only a single surface is created. If the middle button is used, additional surfaces of same type can be made in succession.

2. Surface Classes. The majority of MCNP surfaces are available, and some additional types have been introduced as special cases of a more general type. Planes perpendicular to the cartesian axes (*PX*, *PY*, *PZ*) are available, but the general plane (*P*) cannot yet be created. Planes parallel to the axes (*PYZ*, *PXZ*, *PXY*) have been introduced (these are expressed as a general plane when an MCNP input file is created). Spheres may be centered at the origin (*SO*), have their center constrained to lie on an axis (*SX*, *SY*, *SZ*), or suffer no constraints (*S*). When a general sphere is created, its center is at the depth of the slice in which it is created. Cylinders may be either centered on an axis (*CX*, *CY*, *CZ*) or parallel to an axis (*C/X*, *C/Y*, *C/Z*). Six analogous flavors of elliptical cylinders are available (these are written to an MCNP input file as special quadratic [*SQ*] surfaces). The initial request for a cone must specify whether it is single- or double-sheeted. The orientation of the single sheet cones follows the position of the mouse with respect to the cone's origin. As with cylinders, cones may be aligned in 6 ways (*KX*, *KY*, *KZ*, *K/X*, *K/Y*, *K/Z*). Elliptical cones are under development. Ellipsoids are another separate realization of the special quadratic. As with spheres, ellipsoids may be general or have their center constrained to lie on an axis or at the origin. The initial creation of an ellipsoid in a slice only determines 2 of the 3 radii. The third radius is (arbitrarily) set equal to one of the other two; it can be changed by resizing the ellipsoid in an orthogonal slice or by typing in the *Surface Parameters* popup. Elliptical torii must have their major axis parallel to an axis (*TX*, *TY*, *TZ*). They are created in the slice perpendicular to the major axis by positioning the outer major radius and then the concentric inner major radius. The cross section of the minor radii may be elliptical. When created, this cross section is circular, but may be adjusted by resizing or by typing the values

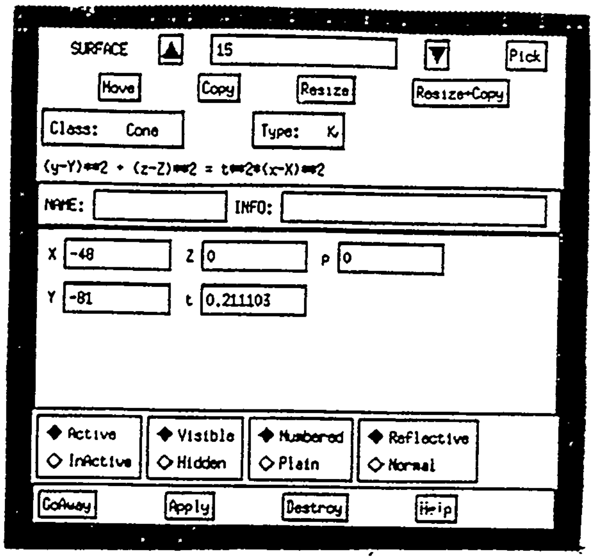

Fig. 4—The Surface Parameters Popup

for the minor radii in the property sheet. Additional specializations of the quadratic surfaces, such as hyperboloids and paraboloids, aligned with one of the axes, are planned, as are tabular surfaces of revolution.

3. Surface Properties. The *Surface Parameters* popup window, show in Figure 4, shows information about a surface and permits changes to the surface and its associated properties. The surface described on the property sheet is the last surface to be created or modified. A different surface may be chosen by typing its ID number, clicking up and down arrows to step through the list of surfaces, or by *Pick*ing with the mouse—the surface closest to the position of the cursor and actually intersecting the slice when the mouse is clicked is the selected surface. The selected surface is highlighted by being redrawn with a thick line (any existing highlighted object is redrawn with the default line thickness). The *Class* (eg., *Torus*) and *Type* (eg., *TX*) of the surface and the defining *Equation* are shown. A *NAME* and an *INFO*rmation field are available for the user to supply these entries (they are displayed if they exist). The coefficients in the surface equation are shown, and may be changed. At present, there is no checking for consistency among the typed coefficients—a negative radius is happily accepted, for example. A surface may be *Hidden*—made invisible, in which case it is not drawn and not available for picking, but will be written to an MCNP input file. A surface may be made *InActive*, which is like being invisible, but will not be written to an MCNP file. An inactive surface is still retained in memory, and may be reactivated. A surface may also be *Destroy*ed, in which case it is gone forever. Each surface carries a flag for whether or not to be numbered (a small number appears next to each numbered surface). A setting of *Plain* (not *Numbered*) overrides the global *NUMBER Surfaces* setting (but a setting of *Numbered* will not override a global setting of not to number surfaces). Any change to a surface attribute (except for *Destroy*) takes effect when the *Apply* button is pressed. Surface creation must be initiated from the main menu: it cannot be done from the *Parameters* popup

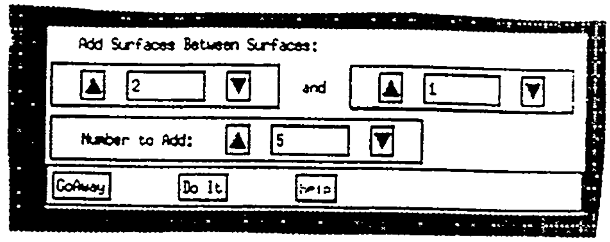

Fig. 5—The AddSurfaces Popup.

4. Surface Modification. Surfaces may be *Move*d or *Resize*d. The resulting surface may be either the original or a new *Copy*. The *Copy* button makes a translated copy of the same size. Inappropriate actions, such as a request to move a sphere centered at the origin, will generate a warning. Both moving and resizing a surface, such as changing the origin and radius of a sphere, requires two operations. The intersection with all slices is shown as the surface is being moved. As a circle is moved in the XY slice across the depth of the XZ slice, for example, a small circle appears in the XZ slice, grows to the full radius of the circle, shrinks, and finally vanishes. The effect of a resize on a torus depends on how the torus is selected. In the slice perpendicular to the axis of the torus, the proximity of the initial mouse click determines whether the inner or outer major radius will be changed. The two radii are not allowed to cross. A resize in an orthogonal slice changes the minor radii.

5. Adding Surfaces. A chosen number of surfaces may be *Add*ed between 2 surfaces of the same type. This procedure is controlled by the *AddSurfaces* popup, which is shown in Figure 5. The coefficients of the new surfaces are linearly interpolated between the originals. The added surfaces are useful for subdividing a region to allow for a gradation in importances. In Figure 3, ellipsoids have been *Add*ed between the innermost and outer surfaces.

D. Bodies

1. Body Classes. Several primative solid bodies may be created in XSM. A rectangular parallelepiped (*RPP*) has all 6 sides parallel to two of the coordinate axes. A *BOX* is a rectangular solid with ar-

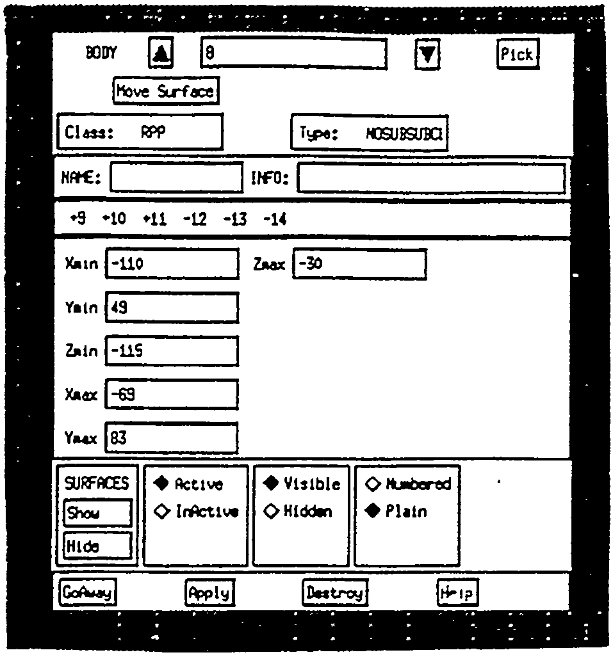

Fig. 6—The Body Parameters Popup.

face not constrained to be aligned with the axes. In XSM, this face must be perpendicular to one of the axes. A right angle wedge (*WED*) is a five-sided solid with one face forming a right triangle (as with the *BOX*, XSM constrains this face to be perpendicular to an axis). Right circular cylinders (*RCC*) and truncated right cones (*TRC*) are the cylinder and cone surfaces bounded by planes perpendicular to their axis. As with the surfaces, the *RCC* and *TRC* solids are either along or parallel to one of the cartesian axes. Spheres, ellipsoids, and torii may be treated as bodies instead of surfaces.

2. Body Creation. The *RPP*, *BOX*, *WED*, and *RCC* are created by dragging the rectangular, triangular, or circular face in one slice. The remaining 2 faces are then positioned in another slice. The original face need not remain in intersection with the slice in which it was defined. A *TRC* is made by first creating a double sheeted cone and then positioning the 2 perpendicular planes in the same slice. Once the first plane is positioned, the second is constrained to lie on the same sheet. Spheres, ellipsoids, and torii are created in the same manner as the corresponding surfaces.

3. Body Properties. The *Body Parameters* popup window, shown in Figure 6, is similar to the *Surface Parameters* popup. It shows the last body made or the body selected by choosing an ID number or *Pick*ing with the mouse. The *Class* and *Type* are shown, and *NAME* and *INFO*rmation fields are available. The body coefficients, typically in the form of an origin, offset

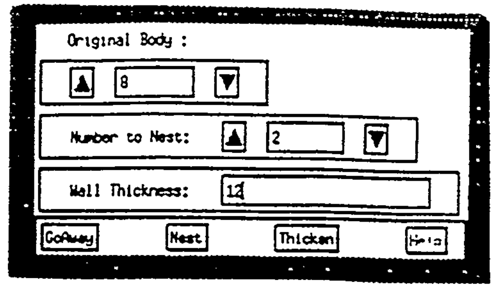

Fig. 7—The NestBodies Popup.

vectors, and radii for bodies other than an *RPP*, are displayed and may be modified. There is not yet any checking of any changed values—for example, the user may change a *BOX* so that some angle is not a right angle. Bodes may be *Hidden* (made invisible) and *InActive*, and carry a *Numbered* flag. Each body is decomposed into surfaces; these surfaces may be *Shown* or *Hidden*. The *Destroy* button will send the body into oblivion.

4. Nesting Bodies. Similar bodies may be *Nest*ed inside an original body. This is controlled by the *Nest Bodies* popup window, which is shown in Figure 7. The dimensions of the chosen number of additional bodies are chosen by linear interpolation between the original body and 0. A single additional body may be created a specified distance away from the original, either outside or inside (negative distance). This ability provides a quick way to create an object with a wall *thickness*.

E. Cells

Cells are described in terms of Boolean combinations of signed surfaces. The simplest cell description, and the only type currently available, is a list of signed surfaces. The intersection of a cell with a slice can be shown as a filled region, and/or the boundary, or outline, of a cell can be shown as a thick line. The union operator is not yet supported. Until the union operator can be used, including bodies in cell descriptions is not feasible.

1. Cell Creation. Cells are created using the *Cell Creation* popup window, which is shown in Figure 8. The first step is to *Pick* the *SURFACES* which will bound the cell. The chosen surfaces are highlighted. When the *SENSE* mode is chosen, a click(s) at a point defines a cell which has the same sense with respect to the chosen surfaces. Surfaces can picked in multiple slices, as is usually necessary to define a closed cell bounded by infinite surfaces. The sense with respect to a surface can only be defined in the slice in which the surface was picked for the cell list. The *Cell Creation* popup

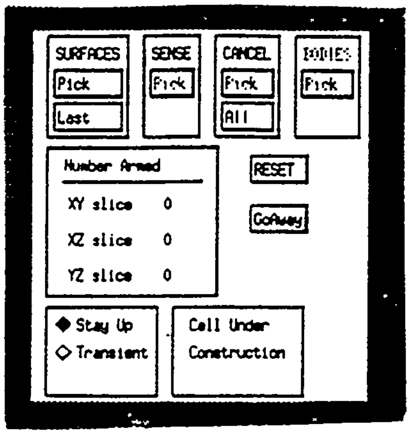

Fig. 8—The Cell Creation Popup.

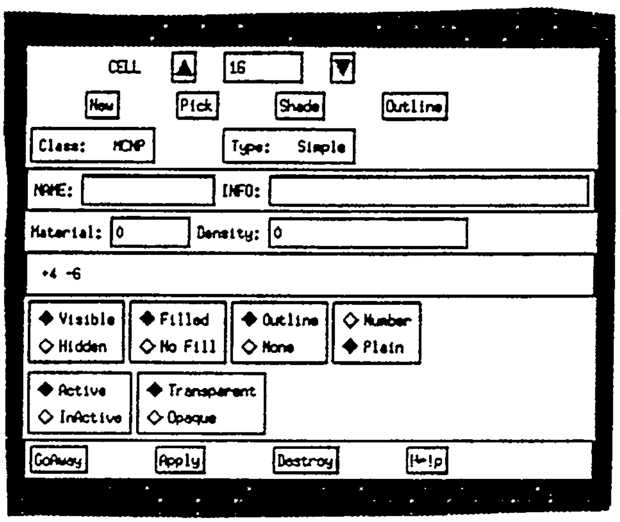

Fig. 9—The Cell Parameters Popup.

shows how many surfaces are *armed* for sense picking in each slice. Before the sense mode is entered, a *Pick*ed surface may be deleted (or *CANCEL*ed) from the list, or *All* picked surfaces may be *CANCEL*ed. The set of surfaces used for the creation of the *Last* cell can be recalled and used for a new cell. This list of last surfaces is (incorrectly) lost when the *Transient* mode is chosen for the *Cell creation* popup, in which mode it goes away following selection of the sense-picking mode.

The *ON SENSE* menu in the main XSM window determines how XSM will respond once a cell has been created. The nonexclusive options include *ReDraw*ing all the slices with the redraw options then in effect, *Shading* only the new cell, *Outlining* the new cell, and doing nothing.

2. Cell Properties. Cell attributes may be modified in the *Cell Parameters* popup window, which is shown in Figure 9. As with the *Surface* and *Body* popups, the cell for which information is displayed is

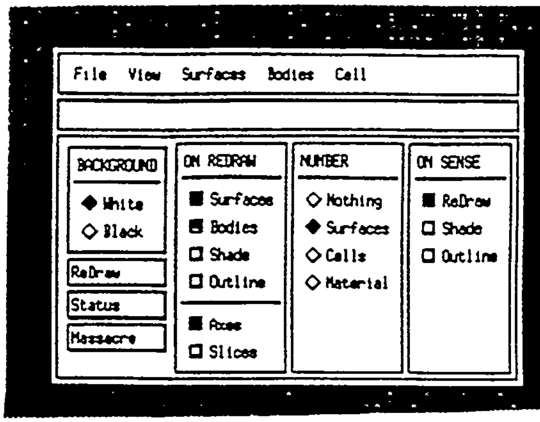

Fig. 10—The XSM main window.

the last cell created, and the selection may be changed by graphically *Pick*ing a cell, by entering an ID number, or by stepping up or down through the cell IDs. The user may point to a cell and have it *Shade*d or *Outline*d. *NAME* and *INFO*rmation fields are available to display and enter those quantities. The *Material* number and *Density* may be entered for the selected cell. The *Cell Description*, in terms of signed surface numbers, is displayed, but cannot be changed. Like surfaces and bodies, cells carry *Active/InActive*, *Visible/Hidden*, and *Number/Plain* settings. *No Fill* and *Outline–None* settings will override the global cell *Shade* and cell *Outline* modes. The cells carry a *Transparent/Opaque* setting which determines whether they will be visible when rendered by Sabrina.

F. XSM Options and Display Settings

A large number of settings control the appearance of the slices and what is drawn in them. The main XSM window, shown in Figure 10, contains a menu bar, an instruction field, and several menus to set graphics preferences. Selecting an entry on the menu bar brings brings up a menu with items which may contain submenus.

1. File Menu. In addition to terminating XSM (the *Quit* option), the *File* menu is used to *Write* either an MCNP or Sabrina input file. At present, this input is only written to the transcript window; in the future the user will be able to instruct XSM to write a file. The Sabrina input consists of the Sabrina MCNP command followed by the MCNP geometry input and Sabrina commands to set the transparency of cells. At least one cell must be nontransparent. This cell attribute is set on the *Cell Parameters* popup. The Sabrina window command, which sets the extent of the volume rendered and is required, is written if a bounding box for the geometry can be computed. Otherwise, the Sabrina file contains a comment that the window command must

be supplied.

2. View Menu. The interactive *ZOOM* is invoked from the *View* menu. The *Slice Depth* entry under *View* is used to position the depth of a slice by dragging the depth in an orthogonal view. The *Test Intersect* function invokes a tracking routine to draw the surfaces. Since the tracking uses the actual equations for the surfaces, this mode has been used during development to check the validity of the surface drawing by speedier algorithms.

3. Object Menus. The *Properties* entry under the *Surfaces*, *Bodies*, and *Cell* menu items brings up the corresponding *Parameters* popup window. Other popups which can be invoked from the main menu are those for *Adding Surfaces*, *Nesting Bodies* (which includes the *Thicken* option), and *Creating* a *Cell*. A *Move* or *ReSize* can be invoked from the *Surfaces* menu.

The *Surfaces* and *Bodies* menus contain a list of object classes which may be created. Most of these classes have a submenu from which a type must be chosen (a sphere centered on the X axis—*SX*—is a type of the sphere class, for example).

4. Instruction Field. The *Instruction Field* gives hints about what should be done with the mouse when XSM is expecting graphical input. The field will be blank when XSM's appetite for such input is satisfied. The text of warning messages, which also appear in the warning popup, will be repeated here.

5. Main Window Buttons. The main XSM window contains several action buttons. The *ReDraw* button causes all slices to be redrawn. The *Status* button writes out some state variables to the transcript pad; it is primarily useful for the developers. Pressing the *Massacre* button destroys all objects and returns most settings to their initial values.

6. Color. Each new surface or body is assigned a color from a set of 20 colors (XSM simply cycles through the list). These colors are relatively dark and are meant for a white background. The same colors, including black, are used when a black background is chosen. In future versions, an alternative set of colors will be used when a black background is chosen.

7. Redrawing. A slice will be redrawn if its window has been obscured and then uncovered. All slices will be redrawn when the geometry changes or in response to the explicit *ReDraw* button. The kind of objects shown in a redraw event is controlled by the *ON REDRAW* menu. The nonexclusive list of choices includes *Surfaces*, *Bodies*, *Shaded* cells, and *Outlined* cells. Two sets of coordinate arrows may be drawn. The *Axes* arrows always pass through the origin while the *Slices* arrows show the depth of the orthogonal slices.

8. Numbering. Surfaces may be numbered. In the future, cells will be able to be numbered by either their identifier or material number. A per-surface no-number setting overrides the numbering of all surfaces. The position of the number reflects the last mouse position when the surface was created. For linear intersections, the position is taken to be relative to the window. For circular and elliptical intersections, the last position is used to define an angle which, together with the radius in the slice, determines the position. There are still some details to be worked out for surface number positioning.

G. Manipulation of the Graphics Areas

1. Graphics Scrolling. Scroll bars may be used to change the portion of the slice being viewed. Surfaces and bodies, but not cells nor surface numbers. are shown as the slice is being scrolled. The maximum extent of the slice is assumed to be five times the initial window size in both width and height (the user will be able to change the maximum extent in future versions) The size of the scroll bar slider reflects the fraction of the maximum extent which is visible.

The size of the slices may changed with the window manager. The pixel to user unit scaling remains in effect—enlarging the window will not increase the size of the objects displayed. Following a resize, the image is centered on the origin and redrawn and the scroll bars adjusted.

2. Scaling. The default scaling is one pixel per user unit. The scaling may be changed by interactive zooming or by typing new scale factors in the *Slice Parameters* popup. An interactive zoom is invoked by chosing *ZOOM* from the *View* menu and then dragging a frame around the region to be enlarged. The scaling in the vertical (y) and horizontal (x) directions need not be the same. If they are the same, the box being dragged remains square. The scaling in two or all three slices may be locked together so that a new scale is propagated to 1 or 2 other slices.

3. Coordinate Labels. Three coordinates are displayed immediately above the graphics area in each slice. The first two show the position of the cursor in the two coordinates of the slice, and the third shows the

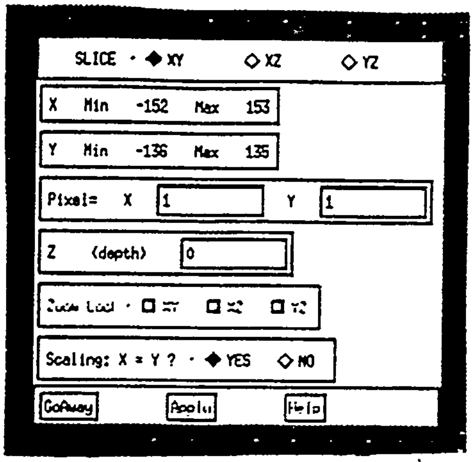

Fig. 11—The Slice Parameters Popup.

depth of the slice.

4. Action Label. Above the coordinate row is a row containing the *Slice Button* and the *Action Label*. The label tells which actions the slice will accept. When a cylinder on the X axis is requested, for example, the *Action Label* for the *YZ* slice will display *CX*, while the other two remain blank. Clicking the mouse in either the *XY* or *XZ* slice will bring up the *Warning Popup* (which must be dismissed by clicking on *OK*).

5. Slice Parameters. The *Slice Button* brings up the *Slice Parameters* popup window, which is shown in Figure 11, or, if that popup is already visible, displays the parameters of the slice whose button is pushed. The menu in the top row of the popup will also change the current slice for which information is displayed. Scrolling or zooming a slice will cause that one to become the current slice. The extent of the slice, in user units, is shown. The scaling factor and the slice depth are shown and may be changed by typing in new numbers. The scaling factors (*Pixel=*) shown are the number of user units per screen pixel. Reducing these values causes the object in the image to be enlarged. If the x and y scaling are locked, which is set by the menu near the bottom of the *Slice Parameters popup*, only 1 value need be entered. The *Zoom Lock* menu determines for which slices the scaling is locked (note that having only 1 slice selected in this menu has no effect).

III. WORK IN PROGRESS

During 1993, the Frankenstein Graphical User Interface (GUI) project consisted of two major efforts, of which XSM was one. The other effort was Igor, which concentrated on decomposing solid body based geometries into polygons for rapid 3 dimensional display. Now that the initial development of these two approaches has been successful, the work will be merged into a single product. This product is yet to be named, but this author favors "Justine". The XSM features and capabilities described here will remain, and the future work planned for XSM will find a home in the merged product.

Additional features that are planned, or that have been started in the Igor code, include a restart capability, reading of existing MCNP input files, and the ability to write input files, at least the geometry portion of them, for other codes in the Los Alamos Radiation Transport Code System (LARTCS). A material definition module is being developed. This module will sort and display cross section files, such as the MCNP XSDIR file. We will also provide GUI based generation of the MCNP data cards; a prototype was worked on this past fall by Jake Anderson as an undergraduate student project.

We believe we are on the verge of producing a powerful and useful input preparation tool for the LARTCS.