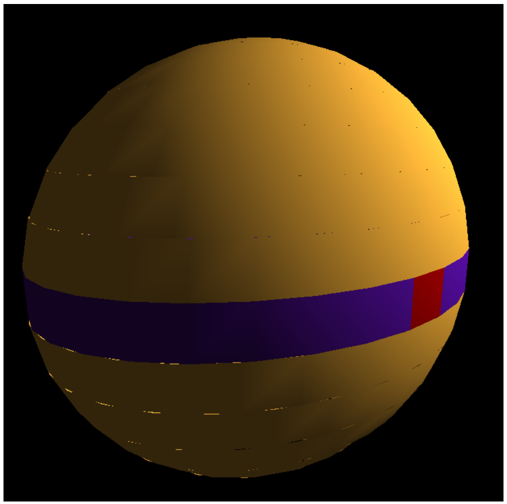
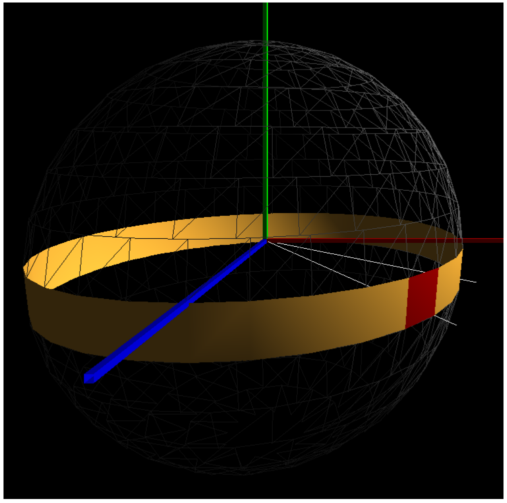
LSU EE 4702-1 **Homework 2** **Due: 8 October 2020**

Problem 0: If necessary, follow the instructions on the `https://www.ece.lsu.edu/koppel/gpup/proc.html` page for account setup and programming homework work flow. Compile and run the homework code unmodified. It should initially show the sphere from the `demo-5-more-ogl.cc` code, but with a *slice* and pair of triangles on the slice, a *segment*, deferentially colored. The segment is red, the rest of the slice is purple. See the upper screenshot to the right. The slice and segment can be moved using the user interface. The goal in this assignment is to provide visual aids to help understand how the sphere is constructed. The sphere's local coordinate axes are to be shown as blocks, and lines to *vertex 2* and to the $\theta = 0$ position on the slice are to be shown. See the lower screenshot to the right.





User Interface

Initially the arrow keys, PageUp, and PageDown can be used to move around the scene. To change η and θ press (lower-case) `s` (if necessary) and then use the arrow keys. The current value is shown in one of the last lines of green text. Pressing `c` toggles between showing the sphere in show-construction mode (the default) and in normal mode.

Press `l` to move the light around and `e` to move the eye (which is what the arrow keys do when the program starts). Pressing `Shift` and an arrow key will move by a larger distance (than if `Shift` were not pressed) and pressing `Ctrl` and an arrow key will move by a smaller distance. The eye can be aimed up and down by pressing `Home` and `End`.

Press `Ctrl=` to increase the size of the green text and `Ctrl-` to decrease the size.

The `+` and `-` keys can be used to change the value of certain variables. These variables specify things such as the light intensity, spring constant, and variables that may be needed for this assignment. The variable currently affected by the `+` and `-` keys is shown in the bottom line of green text. Pressing `Tab` cycles through the different variables.

Code Generation and Debug Support

The compiler generates two versions of the code, `hw02` and `hw02-debug`. Use `hw02` to measure performance, but use `hw02-debug` for debugging. The `hw02-debug` version is compiled with optimization turned off and with OpenGL error checking turned on. You are strongly encouraged to run `hw02-debug` under the GNU debugger, `gdb`. See the material under "Running and Debugging the Assignment" on the course procedures page.

When OpenGL error checking is on (as it is in the debug version) helpful error and warning messages will be printed about misuse or abuse of the OpenGL API. These will appear on the terminal window (which might be a gdb session) from which `hw02-debug` was started.

Keys `y`, `Y`, and `Z` toggle the value of host Boolean variables `opt_tryout1`, `opt_tryout2`, and `opt_tryout3`. The user interface can also be used to modify host floating-point variable `opt_tryoutf` using the `Tab`, `+`, and `-` keys, see the previous section. These variables are intended for debugging and trying things out.

Problem 1: The routine `render()` renders the sphere, and is based on the `demo-5-more-ogl` code. New variables to pay attention to are `opt_show`, `opt_show_eta`, and `opt_show_theta`. When `opt_show` is true the sphere should be shown in construction mode, otherwise it's a plain gold sphere. Also, when `opt_show` is true the lonely triangle should be hidden. Variable `opt_show_eta` identifies the slice which is to be highlighted and `opt_show_theta` emphasizes the segment. Look how these are used to compute `show_eta` and `show_theta`, and how `show_eta` and `show_theta` are used to specify colors. Did you look? This assignment cannot be completed without understanding how these variables are used, and without understanding how the sphere is rendered of the code in render, especially the loop nest.

(*a*) Modify the code so that when in construction mode (`opt_show` is true) the slice is shown in gold (not purple), the segment is shown in red (as it is), and only the edges are shown for triangles which are not part of the highlighted slice; the edges should be gray.

There are several ways to solve this. The least disruptive (in terms of the original code) is to end and start rendering passes within the loop nest, switching between triangles and lines. Do not do more rendering passes than are necessary.

(*b*) Draw two lines that start at the center of the sphere and that pass through points labeled `Vertex 2` in the code. (These are shown as white in the second screenshot on the first page.) The length of these lines should be 1.1 in the sphere's local coordinate space. Both lines should be drawn for points in which `eta0 == opt_show_eta` (for which `show_eta` is true). Draw one line for `theta0 = 0` and draw the other for `theta0 = opt_show_theta`. The second line should pass through a corner of the highlighted segment. Draw these using the `GL_LINES` primitive, and make them white.

Try putting this code after the `slice` / `seg` loop nest.

(*c*) Show the coordinate axes of the sphere local space using red, green, and blue blocks for the x, y, and z axes. The width of the block should be .02 in the sphere local coordinate space.

Set the normals correctly for the blocks. That is, the surface facing up should have a normal of $\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$, etc. If the normals are set correctly then the edges of the blocks will be sharp and the more a block face faces the light the more brightly it will be lit.