Putting points and meshes on the sphere

This is a very open-ended problem that will lend itself to computer experimentation as well as pure math.

Many physical systems are described by differential equations (ordinary or partial). A small fraction of all differential equations have solutions which can be written down in closed form. To get information about the behaviour of solutions to such differential equations computers are used to simulate the equations. Since digital computers deal with discrete objects they cannot deal with differential equations as they stand. Somehow one has to “discretize” the problem and hopefully show that as one makes the discreteness smaller and smaller the solution to the discrete problem approaches that of the real one.

There are many possible ways to “discretize” a continuous system. One way is to replace the derivatives which occur in the equations with the finite differences which occur in the definition of a derivative as a limit. Imagine we have a differential equation with independent variable $x$ and dependent variable $t$

$$\dot{x} = f(x)$$

and wanted to find the solution on the interval $[0, 1]$ when the initial value is $x(0) = a$.

We could do the following: choose an integer $N$ and subdivide $[0, 1]$ into $N$ equal intervals each of length $h = 1/N$, giving us points $N + 1$ points $t_0 = 0, \ldots, t_N = 1$ (the endpoints of the $N$ intervals). Let us call these the gridpoints. At each of the gridpoints we approximate the time derivative of $x$ by

$$\frac{x_{t_{i+1}} - x_{t_i}}{h}.$$ 

Then we replace our differential equation with the difference equation

$$\frac{x_{t_{i+1}} - x_{t_i}}{h} = f(x_i)$$

where $x_i$ is our approximation for the value of $x$ at the gridpoint $t_i$. This equation tells us how to calculate $x_{t_{i+1}}$ if we know the value of $x_i$. Since we know that $x(0) = a$ we take $x_0 = a$ and then can compute all the successive $x_i$.

Under nice circumstances, as we increase the number of gridpoints, we get approximations to the value of $x$ at more and more points, which tend to
the real value. But of course as we take more gridpoints we have to do more computations and the results will be slower. Often we find that solutions to differential equations change fast over some times and more slowly over others. We might like to adapt the distance between successive gridpoints to how fast the solution is changing there. Hopefully this gives us a better approximation to our solution with using so much time or memory. In this way we get a nonuniform grid instead of the previous uniform one.

If instead of ordinary differential equations we look at partial differential equations we might end up looking not at 1-dimensional grids, but multidimensional grids. A uniform example would be to imagine dividing the unit square into $N^2$ identical subsquares. Once again we might want to choose a nonuniform grid – perhaps we divide the unit square up into subsquares of different size.

In problems coming from geometry we often get differential equations which are defined not on a line or a plane but perhaps say on the surface of a sphere (or more general surfaces still). Again we would like to be able to solve these equations numerically and so we need to be able to generate meshes on these surfaces (uniform or nonuniform). This leads to:

**The basic question: find ways to put good meshes on the sphere.**

This question is of course very vague and will depend on what is meant by good. To help you get started you might want to think about some of the following questions.

1. If someone asks you to put $N$ points “evenly distributed” about the sphere how would you do this? What does “evenly distributed” mean? What about approximating the sphere by other geometric objects? Is there a “best” distribution of $N$ points on the sphere? Is there any physical interpretation?

   You might want to start with simple cases. e.g. what about for small numbers of points? what about the simpler cases of points on a line segment, or in a square, or on a circle? Is it easier to distribute evenly for some numbers compared to others?

2. If someone gives you $N$ points on a sphere can you make a mesh out of those points? What would you mean by a mesh on a sphere? What would you mean by a “uniform” mesh on a sphere. What properties might be important in a “good” mesh? What if you are on the plane and not the sphere?
3. If you have some points on the sphere that are not very evenly distributed where should you add points to make the distribution more even? What if you have $N$ evenly distributed points and you want to put more points on the sphere and keep the distribution reasonable even?

4. (Similar to the previous question) What if you have a mesh on the sphere and you decide that it is not fine enough? Is there a nice way to generate a submesh of the original one? What if you want the new mesh to be uniform? What if you want it to be nonuniform e.g. you might only some regions of the sphere where you want to refine the mesh and others where you leave it untouched. Again you might want to think about the circle, the square, etc. and compare with the sphere.

How do you do all these things explicitly on a computer? How does the running time scale with the number of points? Can you do any better?

You should try to find out (at least some of) what is already known about this problem by researching the scientific literature. We will discuss ways to do this in one of the first few class meetings.