My Computer Dreams of Triangles

John Burkardt
Department of Scientific Computing
Florida State University

....

16 April 2016,
Conference for Careers and Undergraduate Research in the Natural Sciences
University of Pittsburgh at Greensburg

....

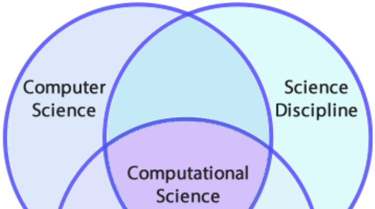
https://people.sc.fsu.edu/∼jburkardt/presentations/. . .
. . . mesh_2016_upg.pdf

Scientific Computing aka Computational Science

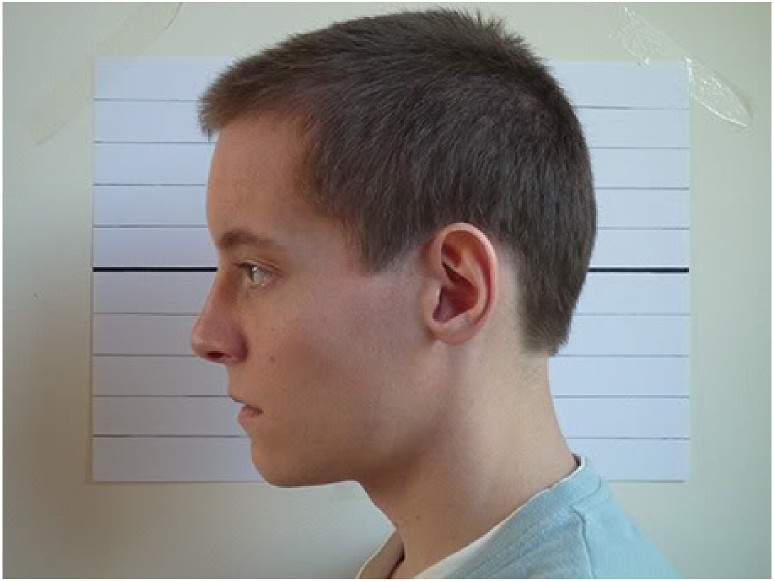
FLORIDA STATE UNIVERSITY

Department of Scientific Computing

sc.fsu.edu

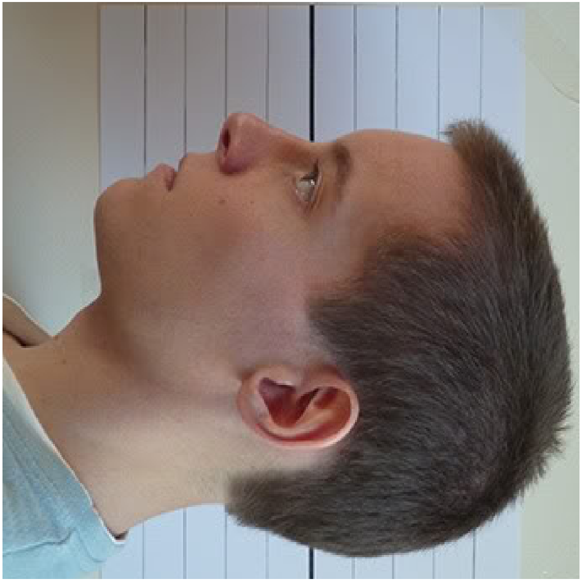


A Profile (Not Mine!)

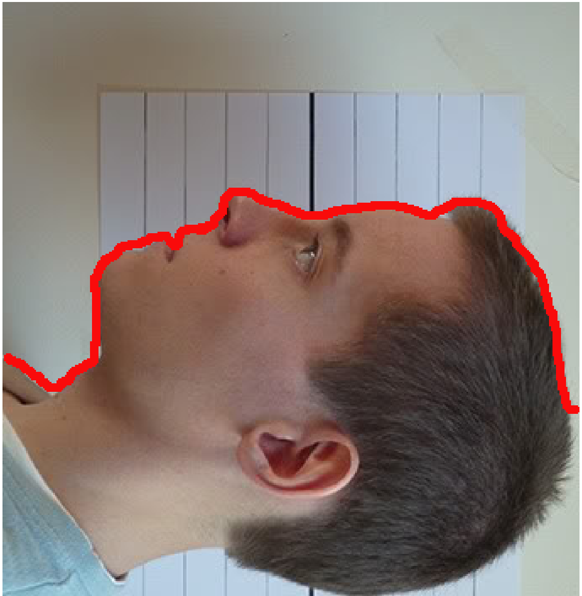


GREENSBURG

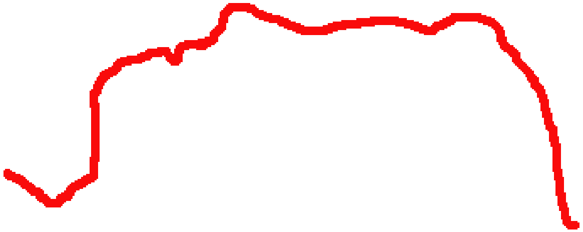
A Profile (Not Mine!) Rotated



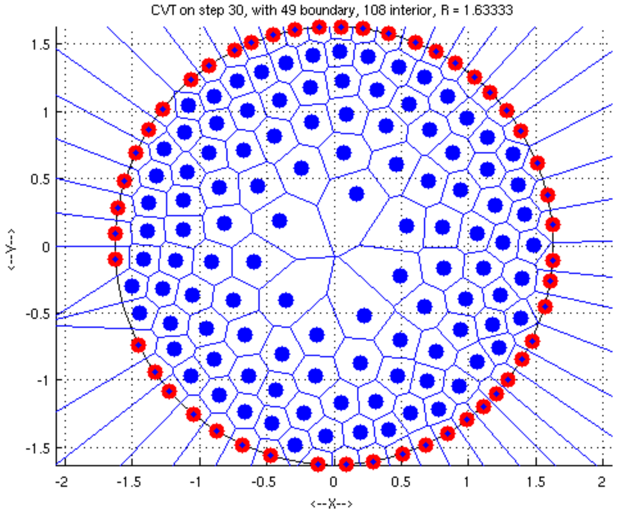
A Profile (Not Mine!) Rotated and Outlined



A Profile (Not Mine!) Rotated and Outlined and Extracted

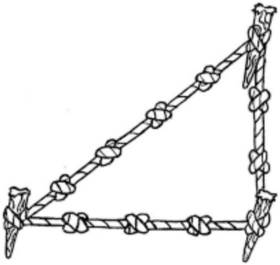
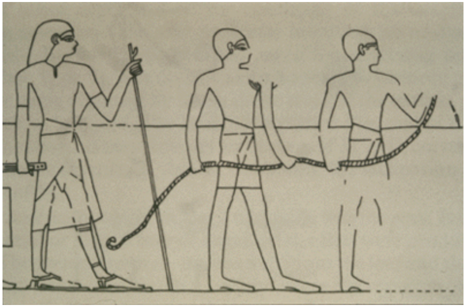


A Flower Grown In a Computer?

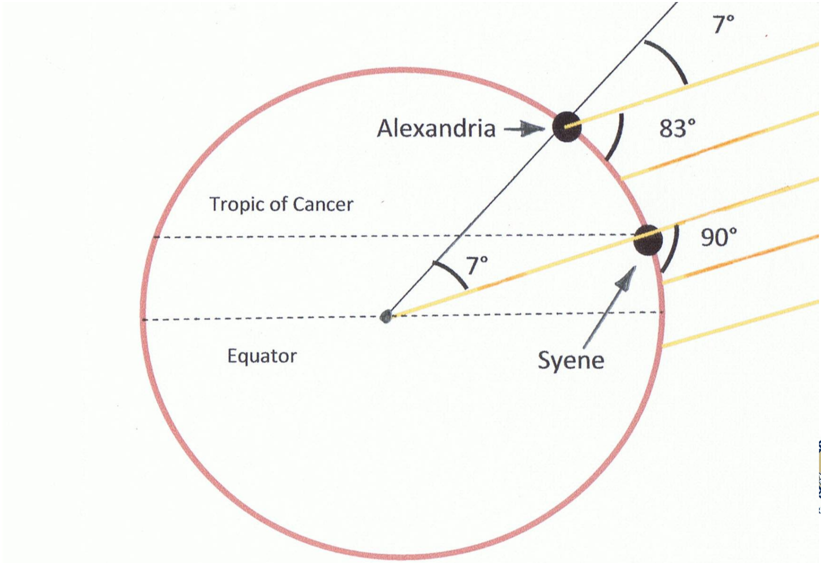


http://people.sc.fsu.edu/∼jburkardt/m_src/cvt_corn/step30.png

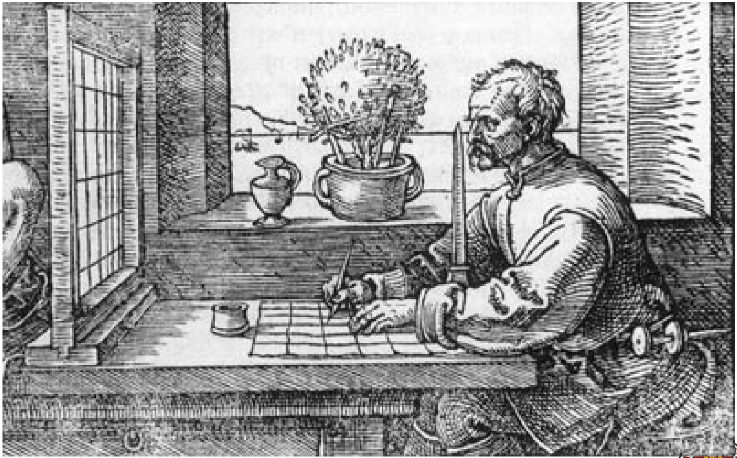
Meshes Started with Taxation

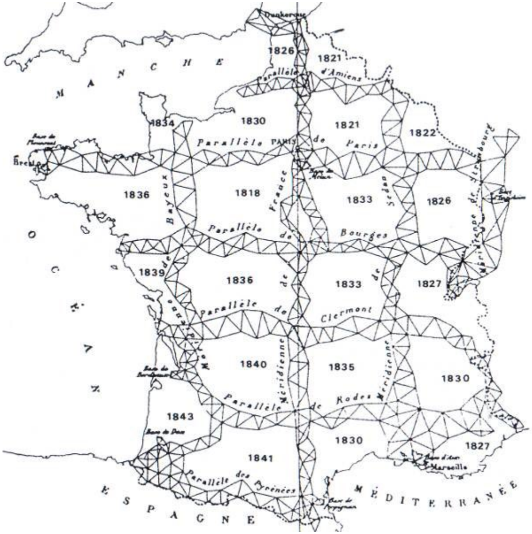
Triangles Measure the Earth



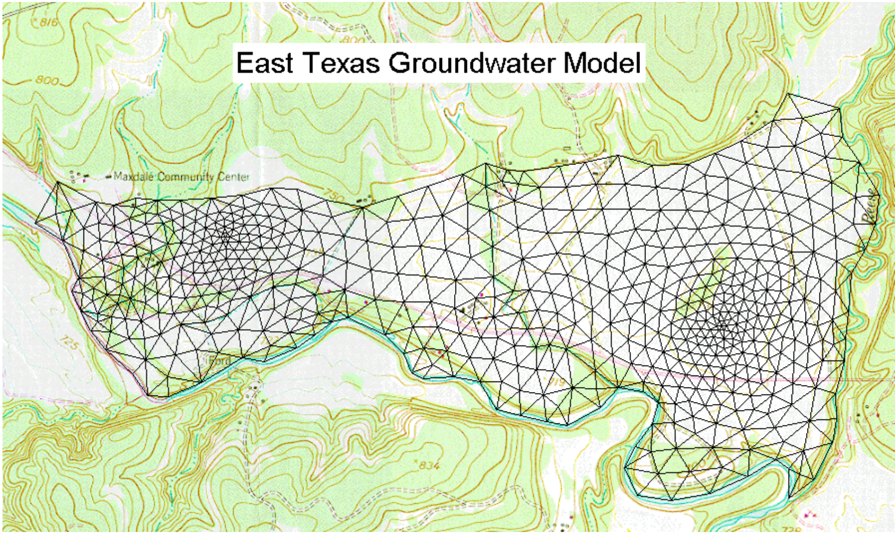
A Mesh Organizes and Simplifies a Complicated View



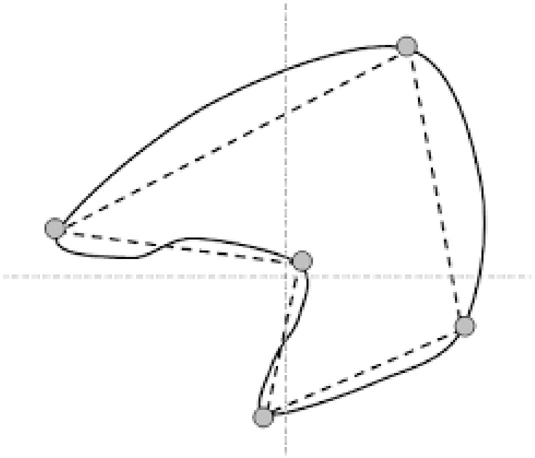
A Mesh Can Organize and Measure Curved Surfaces

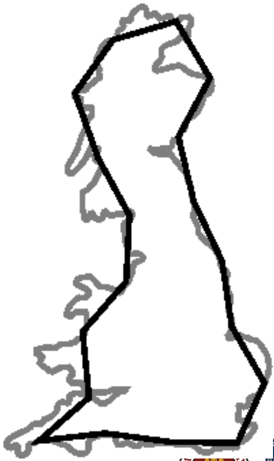


Computers Model Reality with Meshes

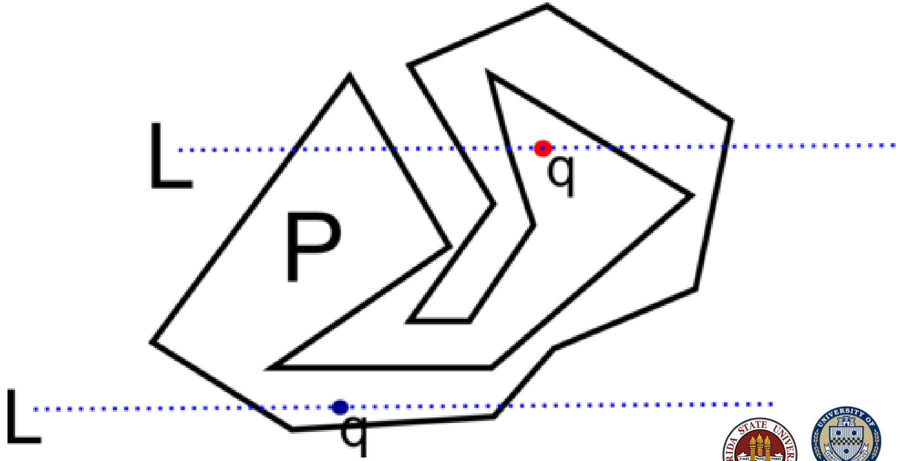


Computers Approximate Shapes with Polygons





Arithmetic is Easy, Geometry is Hard!

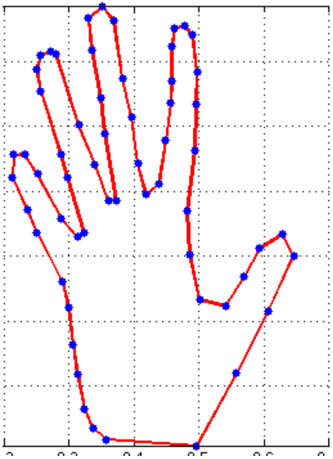


We Can Automate the Measurements

```
%
%  Get the screensize.
%
    screen = get ( 0, 'ScreenSize' );
%
%  Use the whole screen as a figure.
%
  figure ( 'Position', screen );
%
%  Create a graphical coordinate system on the figure.
%
  axes ( 'Position', [ 0, 0, 1, 1 ] );
%
%  The user clicks sample points along the boundary
%  and terminates with RETURN.
%
  [ x, y ] = ginput ( );
```

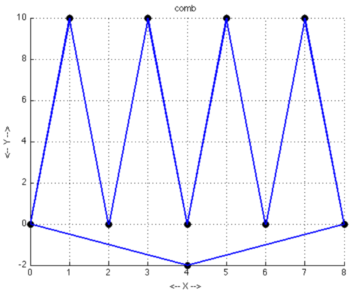
http://people.sc.fsu.edu/∼jburkardt/m_src/hand_data/hand_acquire.m

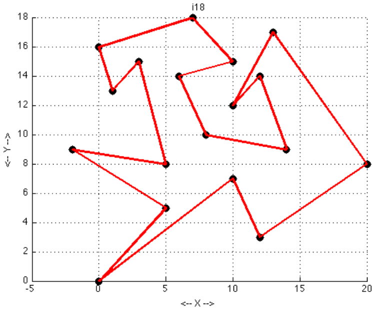
From Points We Have an outline



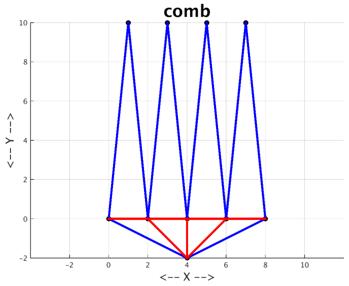
http://people.sc.fsu.edu/∼jburkardt/m_src/hand_data/hand_data.png

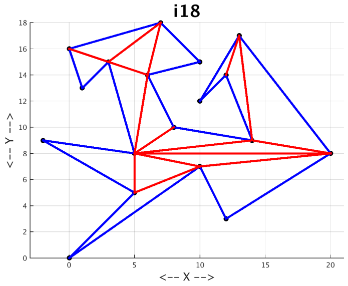
Polygons Are a Little Messy



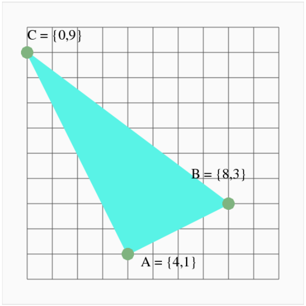


But Polygons Can Be Triangulated





The Area of a Triangle



$$\begin{aligned}
\text{area}(A, B, C) &= 1/2 \begin{vmatrix} B_x - A_x & B_y - A_y \\ C_x - A_x & C_y - A_y \end{vmatrix} \\
&= 1/2((B_x - A_x)(C_y - A_y) - (B_y - A_y)(C_x - A_x)) \\
&= 1/2((8-4)(9-1) - (3-1)(0-4)) \\
&= 20
\end{aligned}$$

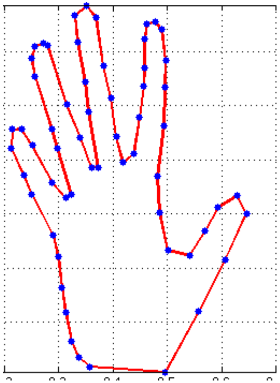
Code for Triangle Area

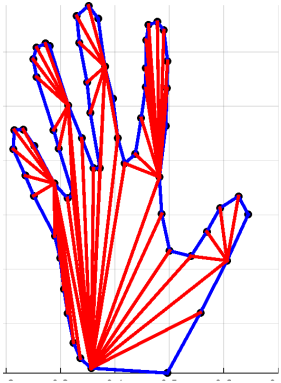
Listing 1: Area of triangle

```
function area = triangle_area ( ax, ay, bx, by, cx, cy )

  area = 0.5 * ...
    ( ( bx - ax ) * ( cy - ay ) ...
    - ( cx - ax ) * ( by - ay ) );

  return
end
```

http://people.sc.fsu.edu/∼jburkardt/m_src/triangle_properties/triangle_area.m

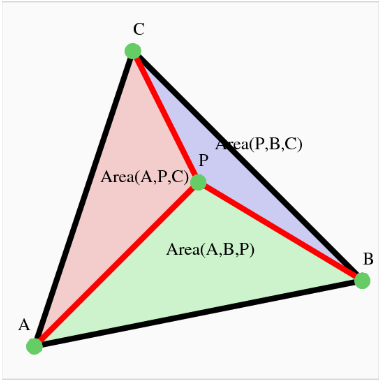
I Can Compute The Area of My (Flat) Hand





http://people.sc.fsu.edu/∼jburkardt/m_src/polygon_triangulate/polygon_triangulate.m

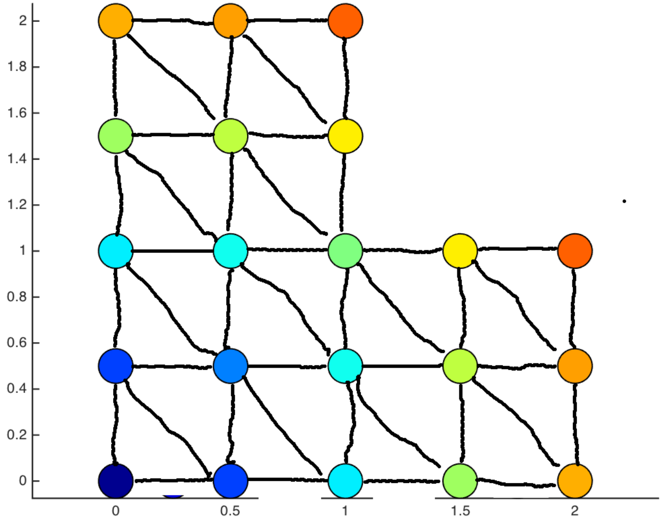
Does a Triangle Contain a Point?



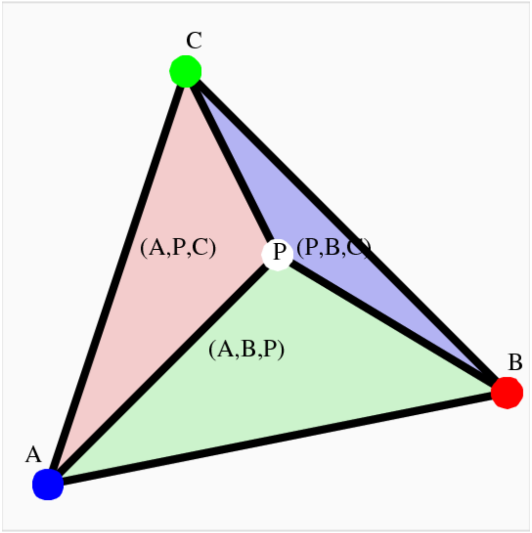
P is inside T if all three areas are positive:

- Area(P,B,C)
- Area(A,P,C)
- Area(A,B,P)

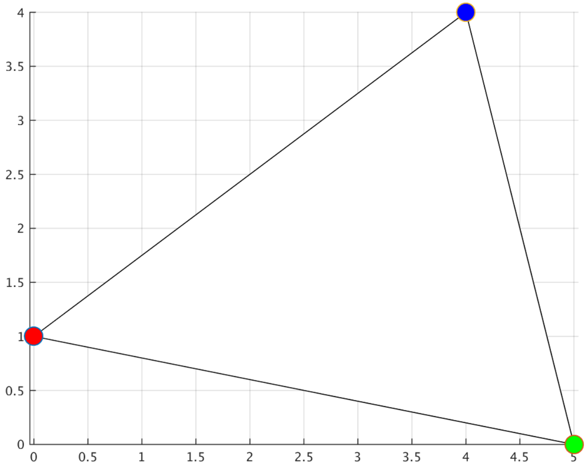
Computer Data Only Computed at a Few Points



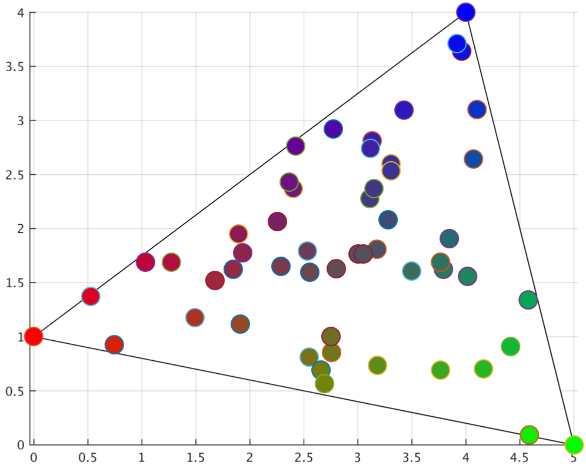
Area-Averaging Vertex Values



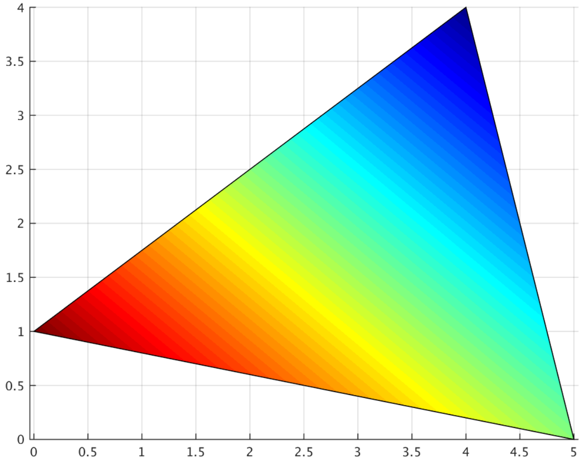
We Know the Values at Triangle Vertices



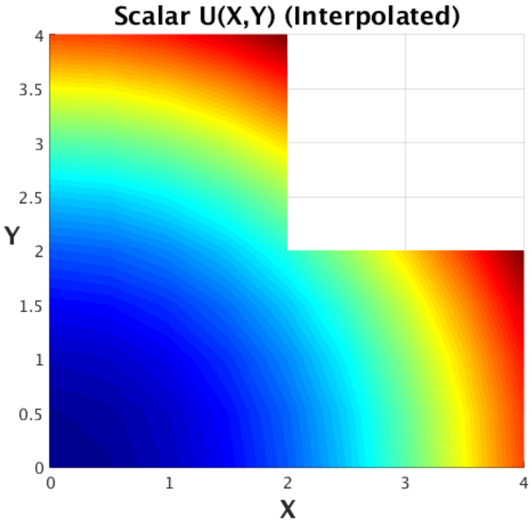
Fill Some Points in the Triangle



Fill All Points in a Triangle



Fill all Triangles in the Region



MESH2D Can Create Internal Nodes

```
[ p, t ] = mesh2d ( v );
```

- *v*, a list of boundary (x,y) coordinates;
- *p*, the coordinates of nodes;
- *t*, triples of nodes forming triangles.

http://www.mathworks.com/matlabcentral/fileexchange/25555-mesh2d-automatic-mesh-generation

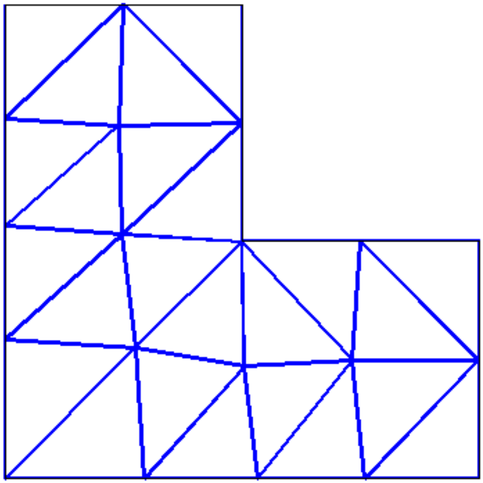
MESH2D: Simple ELL Mesh

```
v = [ 0.0, 0.0; ...
      2.0, 0.0; ...
      2.0, 1.0; ...
      1.0, 1.0; ...
      1.0, 2.0; ...
      0.0, 2.0 ];

[ p, t ] = mesh2d ( v );
```

---

http://people.sc.fsu.edu/∼jburkardt/m_src/mesh2d/ell_demo.m

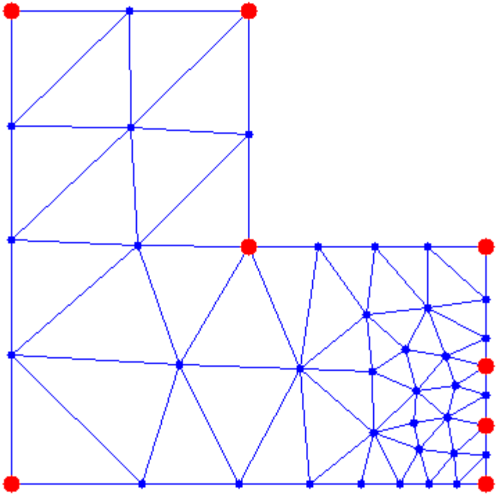
MESH2D: Simple ELL Mesh



MESH2D: Add Two Short Boundary Segments

```
v = [ 0.0, 0.0; ...
      2.0, 0.0; ...
      2.0, 0.25;
      2.0, 0.5; ...
      2.0, 1.0; ...
      1.0, 1.0; ...
      1.0, 2.0; ...
      0.0, 2.0 ];

[ p, t ] = mesh2d ( v );
```

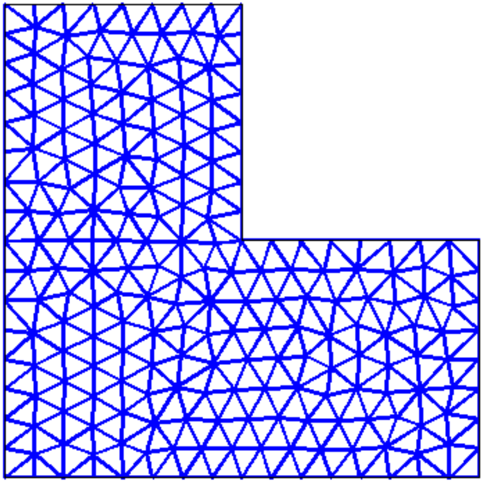
MESH2D: Two Short Boundary Segments



MESH2D: Set Maximum Element Size

```
v = [ 0.0, 0.0; ...
      2.0, 0.0; ...
      2.0, 1.0; ...
      1.0, 1.0; ...
      1.0, 2.0; ...
      0.0, 2.0 ];

hdata = [];
hdata.hmax = 0.1;

[ p, t ] = mesh2d ( v, [], hdata );
```

MESH2D: Set Maximum Element Size



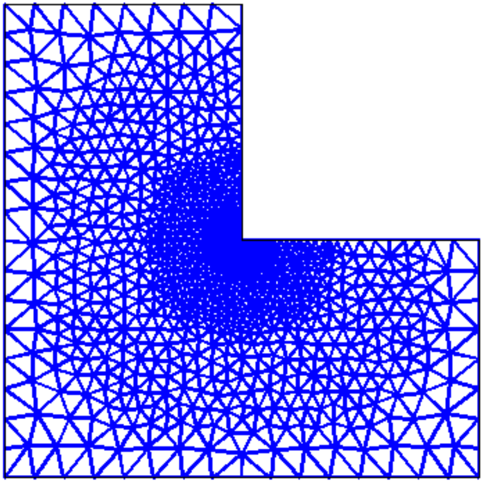
MESH2D: Use a Density Function

```
v = [ 0.0, 0.0; ...
      2.0, 0.0; ...
      2.0, 1.0; ...
      1.0, 1.0; ...
      1.0, 2.0; ...
      0.0, 2.0 ];

hdata = [];
hdata.fun = @hfun;

[ p, t ] = mesh2d ( v, [], hdata );
```

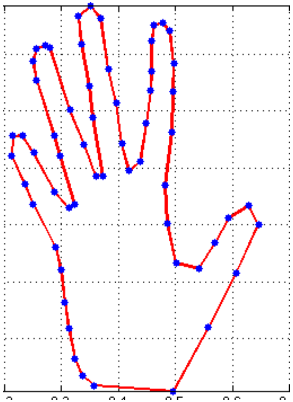
MESH2D: A Density Function Controls Mesh Size H

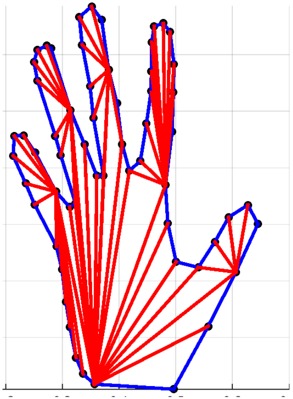
```
function h = hfun ( x, y )

%
%  Minimum size is 0.01, increasing as we move away
%  from ( 1.0, 1.0 ).
%
  h = 0.01 + 0.1 * sqrt ( ( x-1.0 ).^2  + ( y-1.0 ).^2 );

  return
end
```

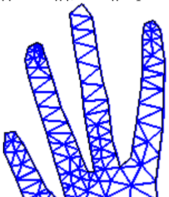
MESH2D: Ask for a Small Mesh Near the Indentation



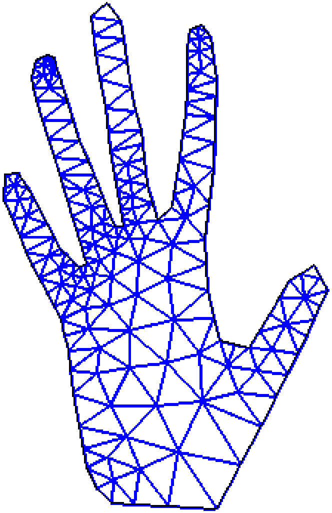
Apply MESH2D to the Hand Data

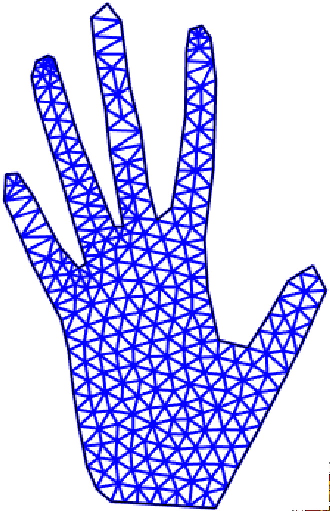






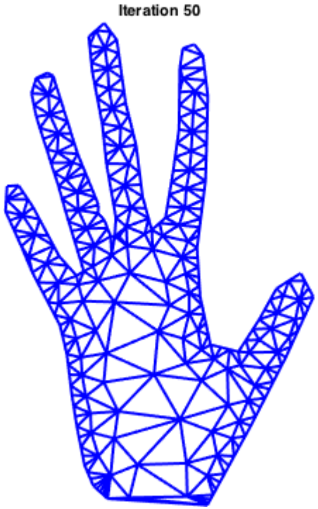
Apply MESH2D to the Hand Data





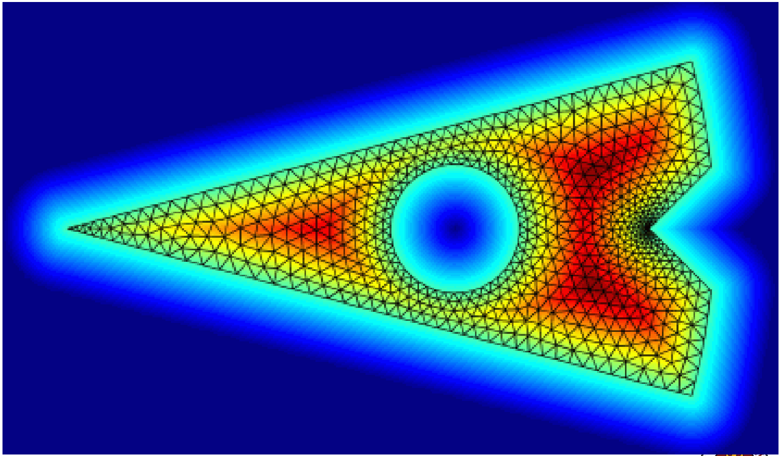
http://people.sc.fsu.edu/∼jburkardt/m_src/mesh2d_hand/mesh2d_hand.m

A Mesh That's Finest Near the Edge



http://people.sc.fsu.edu/∼jburkardt/m_src/distmesh/p24_mesh.png

A Mesh for the Holey Pie

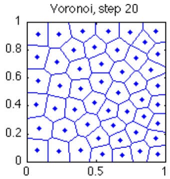


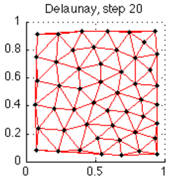
A Mesh for the Holey Pie

Is there time for a quick movie?

http://people.sc.fsu.edu/∼jburkardt/presentations/cvt_movie_p08_cramped.mp4

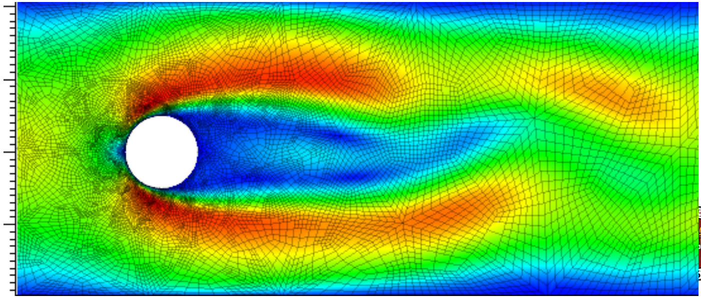
Meshes that Adapt to Flat or Curved Surfaces



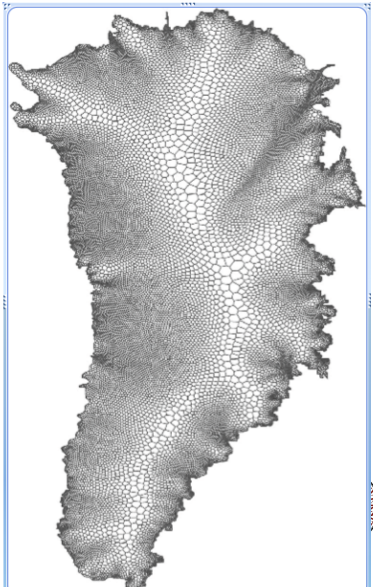


http://people.sc.fsu.edu/∼mgunzburger/

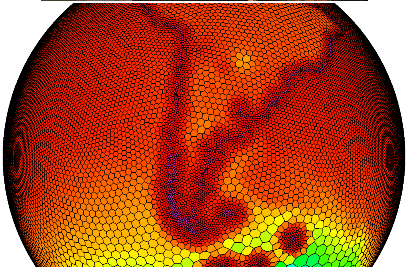
2D: A Moving Mesh for Fluid Flow



2D: A Mesh for Greenland Glaciers



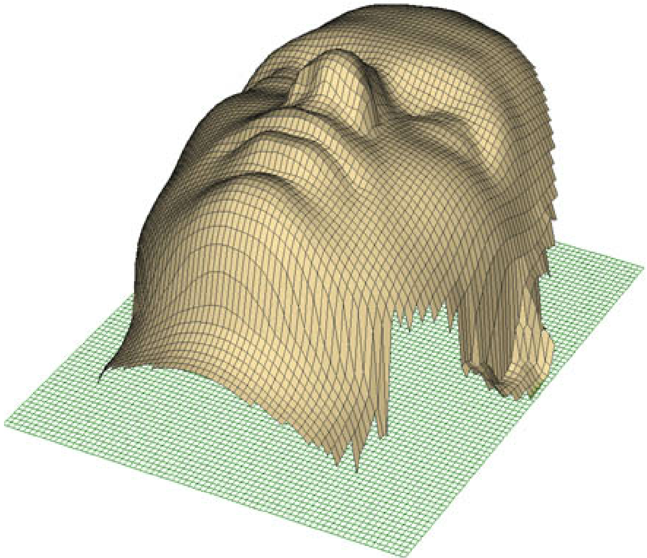
$2\frac{1}{2}$D: Accurate Mesh of Coastline Interfaces



$2\frac{1}{2}$D: Scan, Mesh and Classify Bones for Age-At-Death

$2\frac{1}{2}$D: Scan, Mesh and Classify Arrowheads

$2\frac{1}{2}$D: A Flatbed Scanner Only Gets One Side

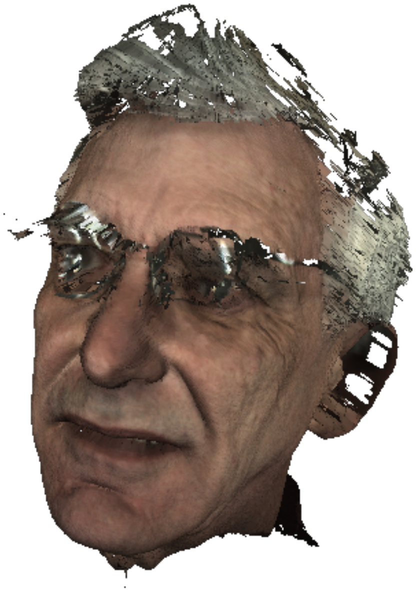


A Wrap-Around Scanner for Head-Protection Studies

http://morphlab.sc.fsu.edu/people/lab_members.html

Results of a Wrap-Around Scan

http://people.sc.fsu.edu/∼jburkardt/presentations/my_head.ply

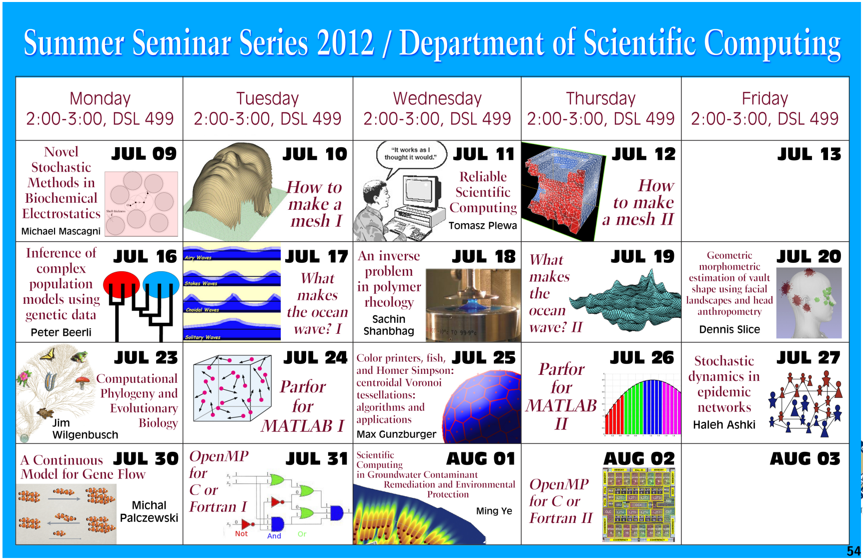


SCANNERS: THE MOVIE

Is there time for a another movie?

http://people.sc.fsu.edu/∼jburkardt/presentations/head_scan_movie.mp4

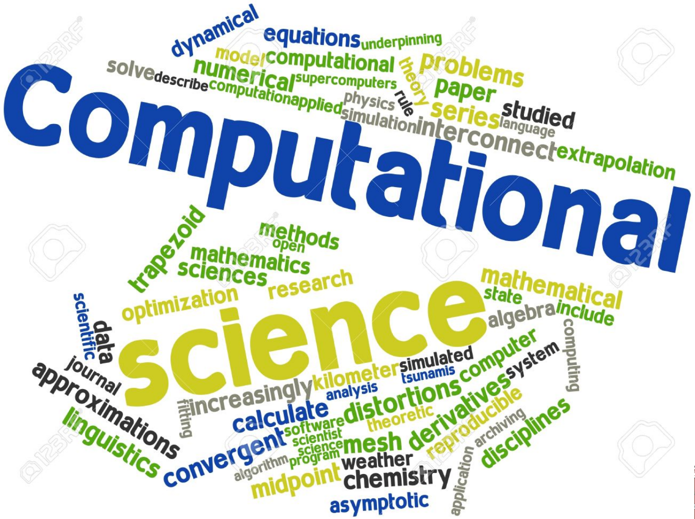
Scientific Computing: A Calendar of Careers...



Summer Seminar Series 2012 / Department of Scientific Computing

Monday 2:00-3:00, DSL 499	Tuesday 2:00-3:00, DSL 499	Wednesday 2:00-3:00, DSL 499	Thursday 2:00-3:00, DSL 499	Friday 2:00-3:00, DSL 499
JUL 09 Novel Stochastic Methods in Biochemical Electrostatics Michael Mascagni	JUL 10 *How to make a mesh I*	JUL 11 "It works as I thought it would." Reliable Scientific Computing Tomasz Plewa	JUL 12 *How to make a mesh II*	JUL 13
JUL 16 Inference of complex population models using genetic data Peter Beerli	JUL 17 Airy Waves Stokes Waves Cnoidal Waves Solitary Waves *What makes the ocean wave? I*	JUL 18 An inverse problem in polymer rheology Sachin Shanbhag	JUL 19 *What makes the ocean wave? II*	JUL 20 Geometric morphometric estimation of vault shape using facial landscapes and head anthropometry Dennis Slice
JUL 23 Computational Phylogeny and Evolutionary Biology Jim Wilgenbusch	JUL 24 *Parfor for MATLAB I*	JUL 25 Color printers, fish, and Homer Simpson: centroidal Voronoi tessellations: algorithms and applications Max Gunzburger	JUL 26 *Parfor for MATLAB II*	JUL 27 Stochastic dynamics in epidemic networks Haleh Ashki
JUL 30 A Continuous Model for Gene Flow Michal Palczewski	JUL 31 *OpenMP for C or Fortran I* Not And Or	AUG 01 Scientific Computing in Groundwater Contaminant Remediation and Environmental Protection Ming Ye	AUG 02 *OpenMP for C or Fortran II*	AUG 03

A New Science, A New Major, New Careers



https://www.siam.org/students/resources/report.php

SLIDE 1/2

Today you will hear about many opportunities for careers in science. I hope I can interest you in looking into my own area, but you may not even have heard of it. Scientific computing, or computational science, has only recently been recognized as its own area, and now degree programs and separate departments are appearing in colleges across the country.

I'd like to say that if you enjoy mathematics, or computer science or any science, but especially enjoy doing these things when there is a computer involved, and an interesting real life problem to solve, then a program in computational science might be worth considering.

Although my first career choice (age 6) was "farmer", my first sensible choice was "something about mathematics", because I loved reading the Mathematical Games articles by Martin Gardner.

Starting in mathematics, my interest in writing, education, programming and research has led me to Scientific Computing, and it's probably too late now to make any more big changes.

SLIDES 3 / 4 / 5 / 6

Mathematics intrigued me because it showed that simple things like numbers and shapes could actually have mysterious properties.

In high school, I read a math book that described a proof that it was possible to cut a ball up into a few pieces that could be reassembled to make two balls of the same volume as the original one.

And I read that equations like $y = f(x)$ could not only describe lines and parabolas and cubics - there was also a formula $y = f(x)$ that corresponded exactly to the profile of my own face.

I decided this was too nonsensical to believe, but I really wanted to hang out with people who thought things up like this!

SLIDE 7

After getting a mathematics degree, I was working at Iowa State University (half time research, half time running the computer lab) when I walked into a graduate student's office and saw a plot on his computer screen that just didn't make sense to me.

It was beautiful and it was mathematical. That doesn't happen often!

My friend, Lili Ju, explained to me that this was actually an example of a kind of mesh that organized itself, almost the way it would happen in nature.

I've got to know more about this! I told him, and I spent some time learning what he had done and how it could be used.

And I think that leads me into my story...

SLIDE 8 / 9 / 10 / 11 /12

The Egyptians used 3-4-5 triangles of rope to resurvey the Nile banks after floods.

The Greeks named geo-metry and trigono-metry, and Eratostenes used this to measure the size of the Earth, the tilt of the Earth's axis, the distance to the sun.

Artists discovered that a complex object could be copied by viewing it through a wire mesh, and then filling in the boxes.

The French revolution invented the meter, and then had to measure a section of longitude in order to figure out what the meter was.

Mathematicians didn't see any reason to divide smooth space up into triangles, but engineers and computational scientists realized the same lesson that artists had discovered: a complex shape may be easier to work with if a regular mesh is used to break it into many similar pieces.

SLIDE 13

It's pretty easy to see how computers can be taught to count, by turning decimal numbers into binary numbers, binary numbers into strings of 0's and 1's, and 0's and 1's into electronic switches.

But how do I explain the shape of Mickey Mouse's head? The area of lake Erie? The branching pattern of a human lung?

Let's take a shape that's not too simple, but not so complex, and "teach" the computer to see it.

Let's lower our expectations, and assume that the shapes we describe will be drawn using a sequence of straight lines. This means we're working with poly-gons (Greek for "many sides").

SLIDE 14

Using a polygon, we can still ask many interesting geometric questions, including

- perimeters,
- areas,
- centers of mass,
- whether a point is inside or outside a shape,
- the distance between a point and a shape
- can I get through a maze?
- how many objects can fit inside this shape?
- do these two shapes fit together?

Answering these questions is sometimes easy for us - but how do we teach the computer to do this?

SLIDE 15 / 16 / 17 / 18 / 19 / 20 / 21

If we have a shape outlined by straight lines, it is always possible to regard it as a collection of triangles.

My hand is an interesting shape (at least to me). Using a program like MATLAB, we can record points on my hand's outline so the computer can "see" what I see.

My hand is represented by a polygon. I don't know a lot about polygons, but I do know a lot about triangles. As it turns out, any polygon can be dissected into triangles, by "slicing off one ear at a time." And you can teach the computer to do this.

Thanks to the Greeks, we know lots of things about triangles, such as the area. There's a formula for the area of any triangle with vertices A, B, C. Using this, I can get the area of (the polygon representing) my hand.

SLIDE 22

It is possible to get a **negative area**. Indeed, if A = (0,0), B = (1,4), C = (3,2), we get area (A,B,C) = -5. Interestingly enough, area (A,C,B) = +5.

The minus sign is telling us something very useful: the triangle (A,B,C) has its vertices listed in clockwise order, but (A,C,B) lists them in counterclockwise order, The sign of the area is a warning about the orientation of the triangle.

As long as we promise to list triangle vertices in counterclockwise order, we will have no problems with the area formula. But it turns out that this bit of knowledge can be used to determine other information.

SLIDE 23

Suppose I have a shape that I have turned into a polygonal outline, and then into a collection of triangles.

And now suppose I ask whether the point P is inside the shape? P is inside the shape (or at least the polygon) if it is inside a triangle. And it is inside triangle ABC exactly if all three subtriangles formed by P have positive area.

So that means we can solve the "point inside shape" geometry problem.

SLIDES 24 / 25 / 26 / 27 / 28

We measure temperature at a few places, but ask for its value anywhere.

To make a good estimate, we need to take the nearest data and somehow spread it to the query point.

If our data is on the vertices of triangles, then for any point P, we can find the triangle containing P, and use the triangle average of the vertex data. That is, the value at P is constructed by using the values at A, B, and C in the proportions of the triangles PBC, APC, and ABP.

Our estimates are most accurate if the triangles are regular shaped and relatively small.

SLIDE 29

Triangulating the hand completely covers the internal area, but it does so with triangles of many different sizes and shapes.

There might be reasons that we want a pattern of triangles that covers the region more evenly in shape and size.

mesh2d is a computer program for which the user only has to describe an outline of the region of interest, that is, a counterclockwise list of points.

We will start by asking for a simple mesh of a simple region, and then push the code little by little to harder tasks.

SLIDES 30 / 31 / 32 / 33 / 34 / 35 / 36 / 37 / 38

Describing the boundary of our region is easy.

MESH2D creates a simple triangular mesh.

If we add two points to the boundary, MESH2D takes the hint and includes more triangles in that area.

We can even specify a maximum triangle size so that MESH2D will fill up the region as we wish.

We can even specify that the triangle size changes in a way that suits us. Here, a related program called DISTMESH makes sure we have very small triangles near the inner corner of the region.

SLIDES 39 / 40 / 41

We can do similar operations on the hand data.

MESH2D creates a simple triangular mesh, with many internal vertices.

We can specify a maximum triangle size of 0.025 inches.

We can specify that triangles should be small near the edges, using a feature that MESH2D provides.

SLIDES 42 / 43

The "computer flower" I saw on Lili Ju's computer screen is an example of a special mesh called a Centroidal Voronoi Tessellation (CVT), a favorite research topic of FSU professor Max Gunzburger.

In the simplest example, you can imagine letting loose a swarm of bees into a small room, and assume every bee wants to avoid the other bees, and the walls, as much as possible. The resulting pattern is a CVT.

We often want meshes that are fine in some areas and coarse in others. That feature can be included in the CVT.

CVT's are computed using iteration; that is, we make an initial guess for the mesh, and then repeatedly improve it. Here's a short movie that suggests how this works.

SLIDES 44 / 45 / 46 / 47 / 48 / 49

Professor Max Gunzburger has investigated the creation and use of these special meshes for regions, surfaces, and volumes.

FSU graduate student Lukas Bystricky studies the flow of fluids past obstacles. Behind the obstacle, the fluid whips up and down, and Lukas makes a mesh that follows the fluid action.

FSU postdoc Mauro Perego (now at Sandia lab) studies the slow flow of the gigantic ice sheet over Greenland. In order to get good results, he had to use detailed information about the coastline, about the height of the land surface below the ice, and about the average ice sheet velocity at each point.

FSU graduate student Geoff Womeldorff (now at Los Alamos lab) was able to create meshes on the land and ocean, with very small elements where land and ocean overlap, in order to do climate studies.

Using this idea, FSU graduate student Detelina Stoyanova was able to estimate the age of skeletons by scanning a particular bone to create a mesh that could be compared against a collection of such scans.

Undergraduates Marcelina Nagales and Alexa Pennavaria have used the

Slides 50 / 51 / 52 / 53 / 54 / 55

FSU Professor Dennis Slice runs a geometric morphology lab. He has always had a 2D scanner, but now he has a 3D head scanner, for research on helmet design. FSU graduate student Alex Townsend sat me down in the barber's chair and promised me it wouldn't hurt.

A laser measures hundreds of thousands of points on the head.

A computer program can then organize these points into a triangular mesh representing the shape of the head.

You may remember that I mentioned at the beginning that mathematics promised me that an equation for the profile of my face "existed", although it offered no way of discovering it. Computational Science has produced a very reasonable result.

It's may be just an approximation, but now I can bet my head on it!