Fields, Meshes, and Interpolation (Part 1)
Looking for notetaker

Hi Everyone,

The AEC is looking for a notetaker for this course. See below.

Thanks!
-Hank

The Accessible Education Center is requesting a peer notetaker for this course. You can earn $25 per credit hour for uploading the notes that you're already taking. If you take clear and comprehensive notes, please go to aec.uoregon.edu to sign up to be a notetaker. You may also enter the CRNs of other classes you are taking to see if there are additional notetaking opportunities.
Outline

• Projects & OH
• Intro
  – SciVis vs InfoVis
  – Very, very basics of computer graphics
• The Data We Will Study
  – Overview
  – Fields
  – Meshes
  – Interpolation
Outline

• Projects & OH
• Intro
  – SciVis vs InfoVis
  – Very, very basics of computer graphics
• The Data We Will Study
  – Overview
  – Fields
  – Meshes
  – Interpolation
Project #1: how’s it going?

- Goal: write a specific image
- Due: “Friday Jan 12th” → “6am Saturday Jan 13th”
- % of grade: 2%
- Goal: get multi-platform issues shaken out ASAP.
- Experience last year was pretty good.

Worth 2% of your grade

Assignment:
1) Download, build, and install VTK.
2) Download and install CMake. Use version 3.X
3) Download the file called data.vtk
4) Make directory called “project1”
5) Download file project1.cxx and CMakeLists.txt from class website and copy them into directory project1
6) Update the VTK_DIR variable in CMakeLists.txt to point to the path of the VTK you just installed.
7) Run CMake. This will create build files.
8) Compile the program. For Unix/Mac, this means “make”
9) Run the program. (How to run is platform dependent … on Linux and Mac, a binary gets generated and you invoke it.)
10) Submit a screenshot of the working program via Canvas
Project #2

• Will assign on Friday
• Don’t want to have to rush through lecture ... will only make things confusing.
Office Hours

• Hank:
  – Tuesday 11-12
  – Thursday 1-230

• Brent
  – Monday 3-4
  – Wednesday 3-4
  – (is this ok?)

• Also: Brent available this week by appointment
Outline

• Projects & OH

• Intro
  – SciVis vs InfoVis
  – Very, very basics of computer graphics

• The Data We Will Study
  – Fields
  – Meshes
  – Interpolation
Scientific Visualization

• An interdisciplinary branch of science
  – primarily concerned with the visualization of three-dimensional phenomena (architectural, meteorological, medical, biological, etc.)
  – the emphasis is on realistic renderings of volumes, surfaces, illumination sources, and so forth, perhaps with a dynamic (time) component.

• It is also considered a branch of computer science that is a subset of computer graphics.

• The purpose of scientific visualization is to graphically illustrate scientific data to enable scientists to understand, illustrate, and glean insight from their data.

Source: wikipedia
Information Visualization

- The study of (interactive) visual representations of abstract data to reinforce human cognition.
  - The abstract data include both numerical and non-numerical data, such as text and geographic information.

Source: wikipedia
Kaela’s question

• Think of data as records
• One piece of data:
  – (x1, x2, x3, x4)
• Example 1:
  – x1 = my age, x2 = my height, x3 = my weight, x4 = my vertical leap
• Example 2:
  – x1 = latitude position, x2 = longitude position, x3 = elevation, x4 = temperature at location (x1, x2, x3)
• If each data record has a location *and* if you want to visualize using that location, then it is scientific visualization.
SciVis vs InfoVis

- “it’s infovis when the spatial representation is chosen, and it’s scivis when the spatial representation is given”
What sorts of data?

Of course, lots of other data too…
What Is Visualization Used For?

• 3 Main Use Cases:
  – Communication
  – Confirmation
  – Exploration
How Visualization Works

• Many visual metaphors for representing data
  – How to choose the right tool from the toolbox?

• This course:
  – Describe the tools
  – Describe the systems that support the tools
Outline

• Projects & OH

• Intro
  – SciVis vs InfoVis
  – Very, very basics of computer graphics

• The Data We Will Study
  – Overview
  – Fields
  – Meshes
  – Interpolation
Computer Graphics

• Defined: pictorial computer output produced, through the use of software, on a display screen, plotter, or printer.
What is computer graphics good for?

• Ed Angel book:
  – Display of information
  – Design
  – Simulation and animation
  – User interfaces
The Basics of Using a Graphics System

• You define:
  – Geometric primitives, typically a triangle mesh
  – Coloring for those geometry primitives
  – A model for a camera (where you are and what you are looking at)

• And then:
  – Communicate this information using an interface (e.g., OpenGL)
  – And something (e.g., GPU) renders the scene

• VTK will do this for us.

• **Visualization**: We will focus on transforming data to geometric primitives.
Outline

• Projects & OH

• Intro
  – SciVis vs InfoVis
  – Very, very basics of computer graphics

• The Data We Will Study
  – Overview
  – Fields
  – Meshes
  – Interpolation
Elements of a Visualization

Legend

Display of Data

Reference Cues

Provenance Information
Elements of a Visualization

What is the value at this location? How do you know?

What data went into making this picture?
Where does temperature data come from?

- Iowa circa 1980s: people phoned in updates

**What is the temperature along the white line?**

6:00pm: Grandma’s friend calls in 82°F.

6:00pm: Grandma calls in 80°F.
What is the temperature at points between Ralston and Glidden?

Distance: D=0, Ralston, IA

Temperature:
- 80°F
- 81°F
- 82°F

Distance: D=10 miles, Glidden, IA
Ways Visualization Can Lie

Visualization Errors:
- Illusion of certainty
- Poor choices of parameters

Data Errors:
- Data collection is inaccurate
- Data collected too sparsely
Outline

• Projects & OH

• Intro
  – SciVis vs InfoVis
  – Very, very basics of computer graphics

• The Data We Will Study
  – Overview
  – Fields
  – Meshes
  – Interpolation
Fields & Spaces

• Fields are defined over “spaces”.
  – We will be considering 2D & 3D spaces.

• Defined by an origin and three vectors (or two vectors) that define orientation.
Scalar Fields

• Defined: associate a scalar with every point in space.
• What is a scalar?
  – A: a real number
• Examples:
  – Temperature
  – Density
  – Pressure

The temperature at 41.2324° N, 98.4160° W is 66°F.

Fields are defined at every location in a space (example space: USA)
• Defined: associate a vector with every point in space.
• What is a vector?
  – A: a direction and a magnitude
• Examples:
  – Velocity

Typically, 2D spaces have 2 components in their vector field, and 3D spaces have 3 components in their vector field.
Vector Fields

Representing dense vector data is hard and requires special techniques.
More fields (discussed later in course)

- Tensor fields
- Functions
- Volume fractions
- Multi-variate data
Outline

• Projects & OH

• Intro
  – SciVis vs InfoVis
  – Very, very basics of computer graphics

• The Data We Will Study
  – Overview
  – Fields
  – Meshes
  – Interpolation
What we want

1. material made of a network of wire or thread.
   "mesh for fishing nets"
   **synonyms:** netting, net, network; **More**

2. an interlaced structure.
   "cell fragments that agglutinate and form intricate meshes"

**verb**

1. (of the teeth of a gearwheel) lock together or be engaged with another gearwheel.
   "one gear meshes with the input gear"
   **synonyms:** engage, connect, lock, interlock; **More**

2. represent (a geometric object) as a set of finite elements for computational analysis or modeling.
An example mesh
An example mesh

Where is the data on this mesh?

(for today, it is at the vertices of the triangles)
An example mesh

Why do you think the triangles change size?
Anatomy of a computational mesh

- Meshes contain:
  - Cells
  - Points

- This mesh contains 3 cells and 13 vertices

- Pseudonyms:
  - Cell == Element == Zone
  - Point == Vertex == Node
Types of Meshes

We will discuss all of these mesh types more later in the course.
Rectilinear meshes

- Rectilinear meshes are easy and compact to specify:
  - Locations of X positions
  - Locations of Y positions
  - 3D: locations of Z positions
- Then: mesh vertices are at the cross product.
- Example:
  - X={0,1,2,3}
  - Y={2,3,5,6}
Rectilinear meshes aren’t just the easiest to deal with ... they are also very common.
Quiz Time

• A 3D rectilinear mesh has:
  – $X = \{1, 3, 5, 7, 9\}$
  – $Y = \{2, 3, 5, 7, 11, 13, 17\}$
  – $Z = \{1, 2, 3, 5, 8, 13, 21, 34, 55\}$

• How many points?  $= 5 \times 7 \times 9 = 315$
• How many cells?  $= 4 \times 6 \times 8 = 192$
Definition: dimensions

- A 3D rectilinear mesh has:
  - $X = \{1, 3, 5, 7, 9\}$
  - $Y = \{2, 3, 5, 7, 11, 13, 17\}$
  - $Z = \{1, 2, 3, 5, 8, 13, 21, 34, 55\}$

- Then its **dimensions** are $5 \times 7 \times 9$
How to Index Points

• Motivation: many algorithms need to iterate over points.

```c
for (int i = 0 ; i < numPoints ; i++)
{
    double *pt = GetPoint(i);
    AnalyzePoint(pt);
}
```
Schemes for indexing points

Logical point indices

0.5 1.5 2.5 3.5 4.5 5.5
0.4 1.4 2.4 3.4 4.4 5.4
0.3 1.3 2.3 3.3 4.3 5.3
0.2 1.2 2.2 3.2 4.2 5.2
0.1 1.1 2.1 3.1 4.1 5.1
0.0 1.0 2.0 3.0 4.0 5.0

Point indices

30 31 32 33 34 35
24 25 26 27 28 29
18 19 20 21 22 23
12 13 14 15 16 17
06 07 08 09 10 11
00 01 02 03 04 05

What would these indices be good for?
How to Index Points

• Problem description: define a bijective function, $F$, between two sets:
  – Set 1: $\{(i,j,k): 0 \leq i < nX, 0 \leq j < nY, 0 \leq k < nZ\}$
  – Set 2: $\{0, 1, ..., n\text{Points}-1\}$

• Set 1 is called “logical indices”
• Set 2 is called “point indices”

Bijective: for every element in set 1, there is an element in set 2. And vice-versa.

Note: for the rest of this presentation, we will focus on 2D rectilinear meshes.
How to Index Points

- Many possible conventions for indexing points and cells.
- Most common variants:
  - X-axis varies most quickly
  - X-axis varies most slowly

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>1.5</td>
<td>2.5</td>
<td>3.5</td>
<td>4.5</td>
<td>5.5</td>
</tr>
<tr>
<td>0.4</td>
<td>1.4</td>
<td>2.4</td>
<td>3.4</td>
<td>4.4</td>
<td>5.4</td>
</tr>
<tr>
<td>0.3</td>
<td>1.3</td>
<td>2.3</td>
<td>3.3</td>
<td>4.3</td>
<td>5.3</td>
</tr>
<tr>
<td>0.2</td>
<td>1.2</td>
<td>2.2</td>
<td>3.2</td>
<td>4.2</td>
<td>5.2</td>
</tr>
<tr>
<td>0.1</td>
<td>1.1</td>
<td>2.1</td>
<td>3.1</td>
<td>4.1</td>
<td>5.1</td>
</tr>
<tr>
<td>0.0</td>
<td>1.0</td>
<td>2.0</td>
<td>3.0</td>
<td>4.0</td>
<td>5.0</td>
</tr>
</tbody>
</table>

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
</tr>
<tr>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
</tr>
<tr>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>06</td>
<td>07</td>
<td>08</td>
<td>09</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>00</td>
<td>01</td>
<td>02</td>
<td>03</td>
<td>04</td>
<td>05</td>
</tr>
</tbody>
</table>
Bijective function for rectilinear meshes for this course

```c
int GetPoint(int i, int j, int nX, int nY)
{
    return j*nX + i;
}
```

![Grid of points](image)
Bijective function for rectilinear meshes for this course

```c
int *GetLogicalPointIndex(int point, int nX, int nY)
{
    int rv[2];
    rv[0] = point % nX;
    rv[1] = (point/nX);
    return rv; // terrible code!!
}
```
int *GetLogicalPointIndex(int point, int nX, int nY)
{
    int rv[2];
    rv[0] = point % nX;
    rv[1] = (point/nX);
    return rv;
}
Quiz Time #2

- A mesh has dimensions 6x8.
- What is the point index for (3,7)? = 45
- What are the logical indices for point 37? = (1,6)

```c
int GetPoint(int i, int j, int nX, int nY)
{
    return j*nX + i;
}
```

```c
int *GetLogicalPointIndex(int point, int nX, int nY)
{
    int rv[2];
    rv[0] = point % nX;
    rv[1] = (point/nX);
    return rv; // terrible code!!
}
```
Quiz Time #3

• A vector field is defined on a mesh with dimensions 100x100
• The vector field is defined with double precision data.
• How many bytes to store this data?

\[ = 100 \times 100 \times 2 \times 8 = 160,000 \]
Bijective function for rectilinear meshes for this course

```c
int GetCell(int i, int j, int nX, int nY)
{
    return j*(nX-1) + i;
}
```
Bijective function for rectilinear meshes for this course

```c
int *GetLogicalCellIndex(int cell, int nX, int nY)
{
    int rv[2];
    rv[0] = cell % (nX-1);
    rv[1] = (cell/(nX-1));
    return rv; // terrible code!!
}
```
Outline

- Projects & OH
- Intro
  - SciVis vs InfoVis
  - Very, very basics of computer graphics
- The Data We Will Study
  - Overview
  - Fields
  - Meshes
  - Interpolation
Goal: have data at some points & want to interpolate data to any location
Linear Interpolation for Scalar Field $F$
Linear Interpolation for Scalar Field \( F \)

- General equation to interpolate:
  - \( F(X) = F(A) + t \cdot (F(B) - F(A)) \)
- \( t \) is proportion of \( X \) between \( A \) and \( B \)
  - \( t = (X-A)/(B-A) \)
Quiz Time #4

• F(3) = 5, F(6) = 11
• What is F(4)?
  \[ F(4) = 5 + \frac{(4-3)}{(6-3)} \times (11-5) = 7 \]

• General equation to interpolate:
  \[ F(X) = F(A) + t \times (F(B) - F(A)) \]
  \[ t = \frac{(X-A)}{(B-A)} \]
Bilinear interpolation for Scalar Field $F$

$F(0,1) = 1$
$F(1,1) = 6$
$F(0.3, 1) = 2.5$
$F(0.3, 0) = 8.5$

What is value of $F(0.3, 0.4)$?

$F(0.3, 0.4) = 6.1$

- General equation to interpolate:
  $F(X) = F(A) + t*(F(B)-F(A))$

Idea: we know how to interpolate along lines. Let’s keep doing that and work our way to the middle.