Data

Lecture 5

Data

- Data is what fills the computational arrays discussed earlier.
- Data in computational arrays has no generic physical mapping.
- Using mesh information we map data into physical space.

Mapping Data onto a Mesh

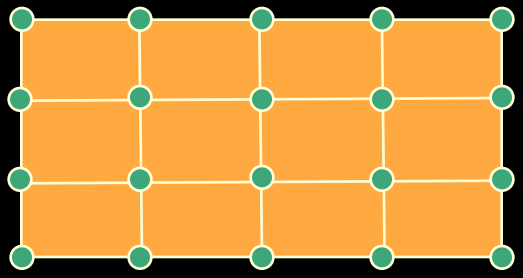
- There is however more than one way to map the data onto a mesh.

- The computational data may be:
  - *Node Data* - data at a point
  - *Cell Data* - data for a finite volume

Node Data

Computational Data Array

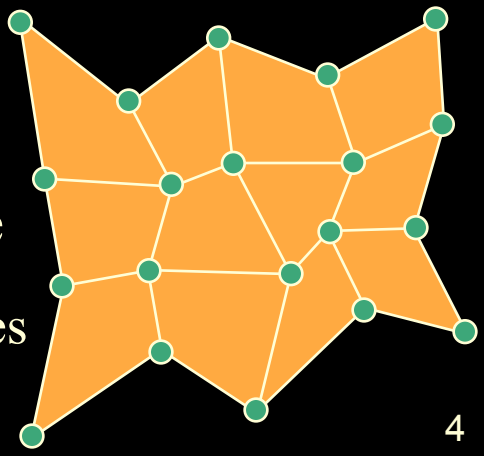
23	41	13	46	19
18	72	42	9	85
45	13	35	26	12
67	34	27	66	11



Physical Mapping

depends on *mesh* type

Here data forms node values

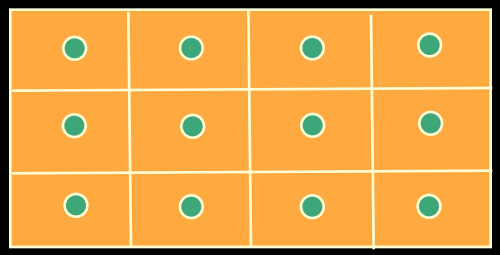


Cell Data

Computational

Data Array

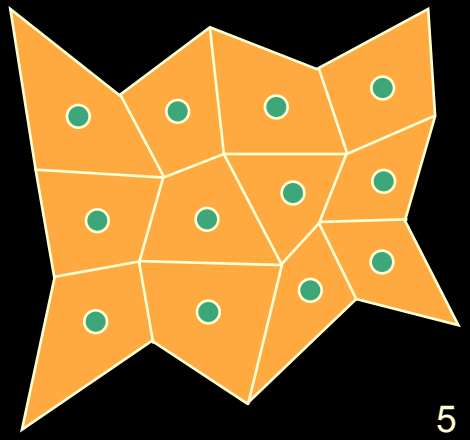
23	41	13	46	
18	72	42	9	
45	13	35	26	



Physical Mapping

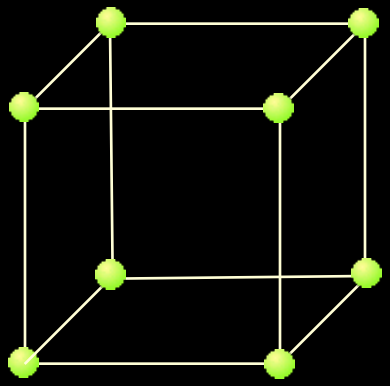
depends on *mesh* type

Here data forms cell values

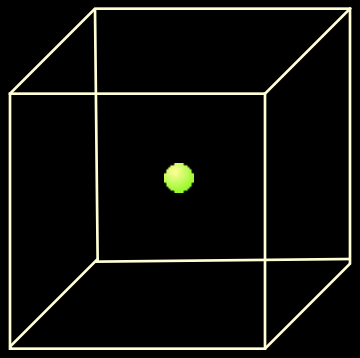


3D Node and Cell Data

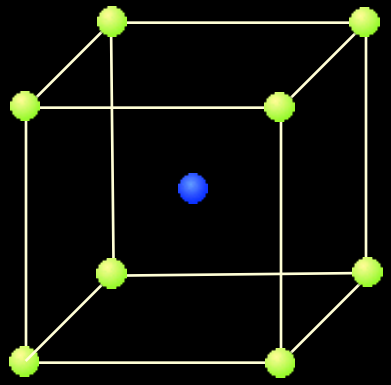
- Node Based



- Cell Based

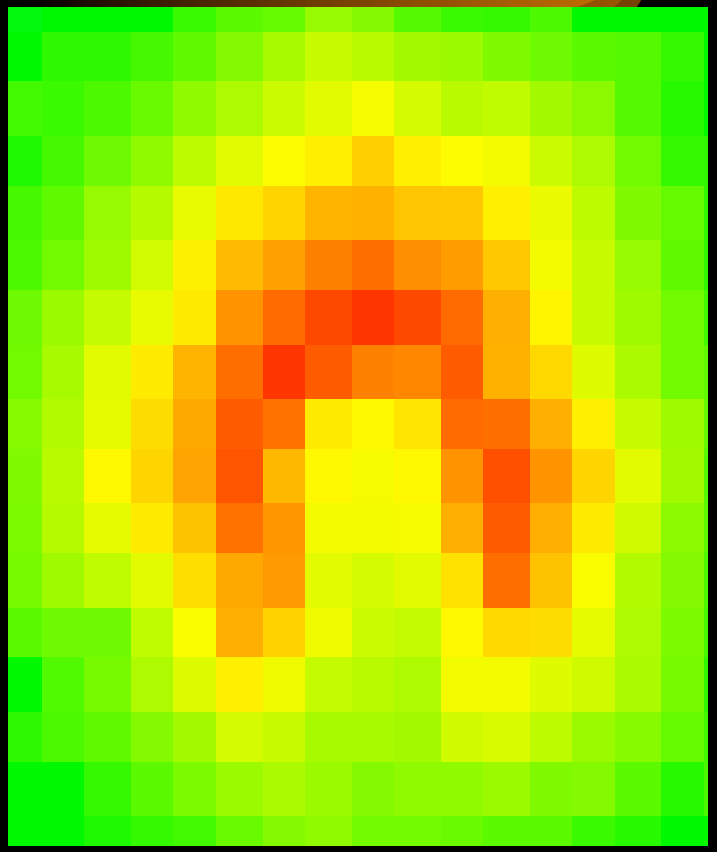


Combination



Visualising Node Data

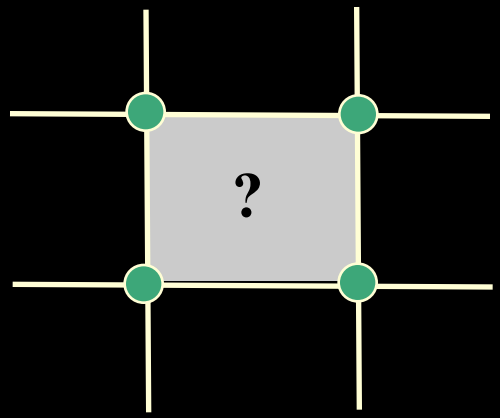
- Node data exists at a mathematical point.
- One technique is to display only coloured points.
- Node data is often displayed as an area.
- Why ? (Isn't this cell data ?)
  - Easier to see
- How does node data map to area ?



Node Data mapped to area

Facet Colouring with Node Data

- What facet colour ?
- There are 4 surrounding node values
- Methods include:
  - Average values
  - Choose corner (eg Bottom Left)
  - Offset grid by 1/2 cell
- Last two are similar
  - They do however spatially extend dataset by a row and column of cells.



Visualising Cell Data

- Two general approaches
  - Using 2D techniques by visualising the external surfaces
  - Using 3D volume visualisation techniques.

Data Types

Data Types

- Two perspectives of numeric data:
  - Abstract Scientific
    - Scalar, Vector
  - Computer Representation
    - Byte, Floats, Integers …

Scalar and Vector Data

- The term *scalar* usually refers to the physical nature of the values.

  Temperature, Pressure, Height, Time are scalars

  Velocity, Gradients, Forces are vectors

- It is possible to have more than one value per node or cell.

Vector Data

- Vector data is normally represented as a set of orthogonal components.
- Some visualisation packages treat vector components just as a set of scalars, leaving it to the user to perform the appropriate operations.

Data Formats

Bits ‘n’ Bytes

Computer Data Types

- Understanding the intricacies of computer data types is an unfortunate necessity in scientific visualisation.
- Why ?
  - Data is stored in various formats
  - Formats have different limitations

Computer Representations

- There are many different numerical formats
  - Integers (8+ formats)
  - Floats (3+ formats)
- Why so many ? Speed/memory/accuracy
  - Float is not necessary more “accurate” than int
  - Some numbers cannot be exactly represented
  - Eg currency is often stored as integers

Integer Representations

- The following ranges are typical but may vary among computer systems
  - Bits
    - 0 or 1
  - Bytes (chars)
    - 0 to 255 (unsigned)
    - -128 to +127 (signed)
  - Shorts
    - 0 to 65,535 (unsigned)
    - -32,768 to 32,767 (signed)

Integer Representations…

- Ints
  - 0 to 4,294,967,295 (unsigned)
  - -2,147,483,648 to 2,147,483,647 (signed)
- Longs
  - 2^{64} -- ie “lots”

Floating-point Representations

- Floats
  - Around 8 digits of precision, powers +/- 38
- Doubles
  - Around 15 digits of precision, powers +/- 308

Storage Formats

Storage formats

- The type is not the end of the story
- So far we have just dealt with the abstract concept.
- Other issues including Binary vs ASCII

ASCII

- Human readable
- Transport across different computer systems
- Takes around 3-5 times more space than binary
- No standard format but easy to interpret
- May not have a column delimiter (some FORTRAN)

Binary

- Inscrutable file formats
- Compact
- Transportation issues
  - Floating-point format
  - Integer formats
  - Big Endian / Little Endian
- Network formats: XDR

End

Lecture 5