Imperial College London

Computational Physics: Interpolation

Yoshi Uchida

Blackett Laboratory

5 January 2018

Welcome to Computational Physics!

(again)

A Bit About Me

Professor Yoshi Uchida, Blackett Room 524

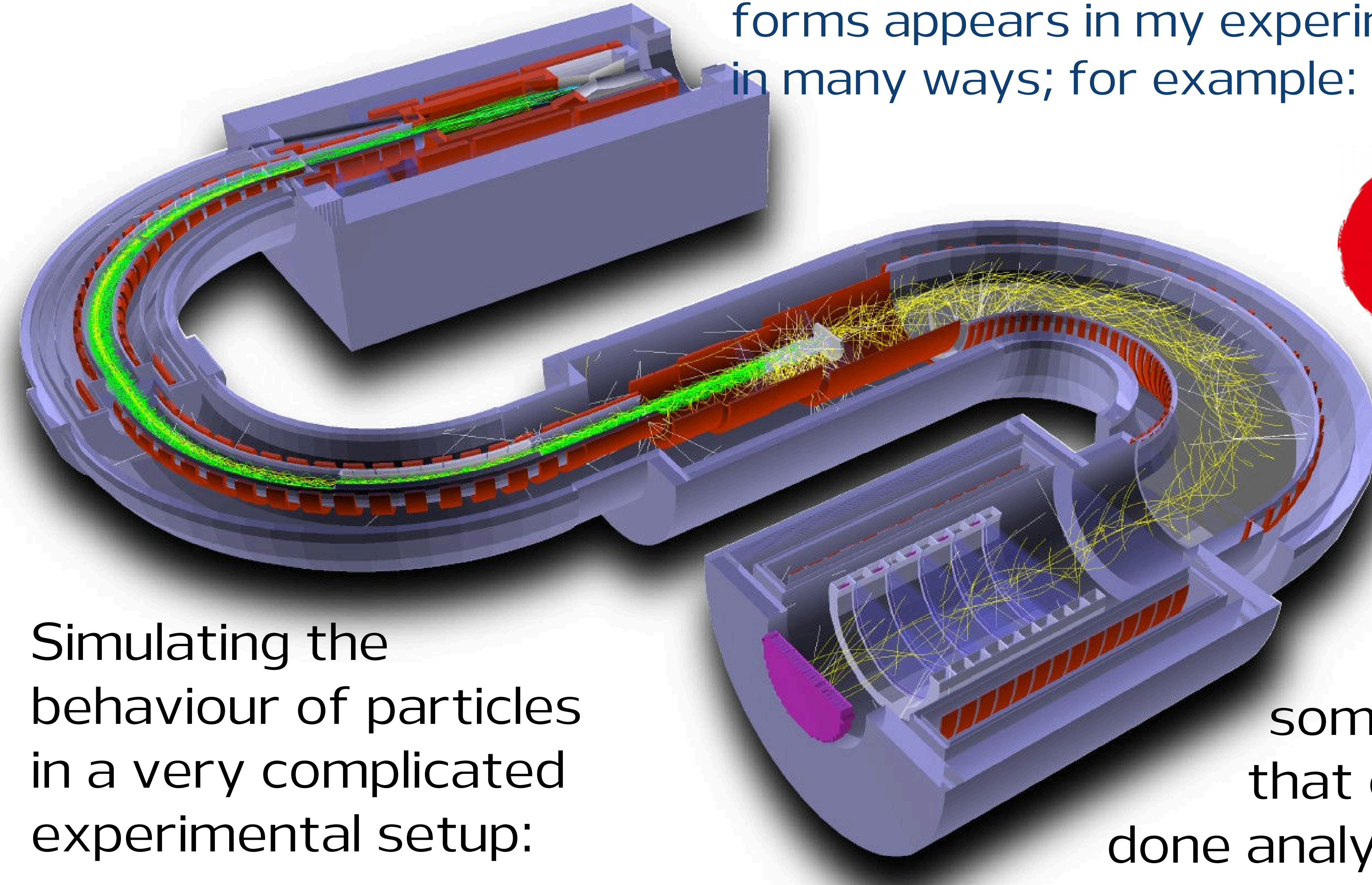
- Particle Physicist (but not a collider person)
  - looking at particle phenomena that are sensitive to the things we don't know about the Universe, without actually trying to *produce* any as-yet unknown particles by brute force
  - mainly neutrinos and muons, for now
- Grew up in Surrey
- PhD from the Massachusetts Institute of Technology
  - when I did work on a collider at CERN
- Postdoc at Stanford University
  - where I moved into neutrinos
- At Imperial College HEP since 2004
  - for more neutrinos, and where I also started looking at muons

CP in Particle Physics

- Usually refers “Charge-and-Parity” symmetry

CP in Particle Physics

- Usually refers "Charge-and-Parity" symmetry
- But Computational Physics in both basic and more advanced forms appears in my experiments in many ways; for example:

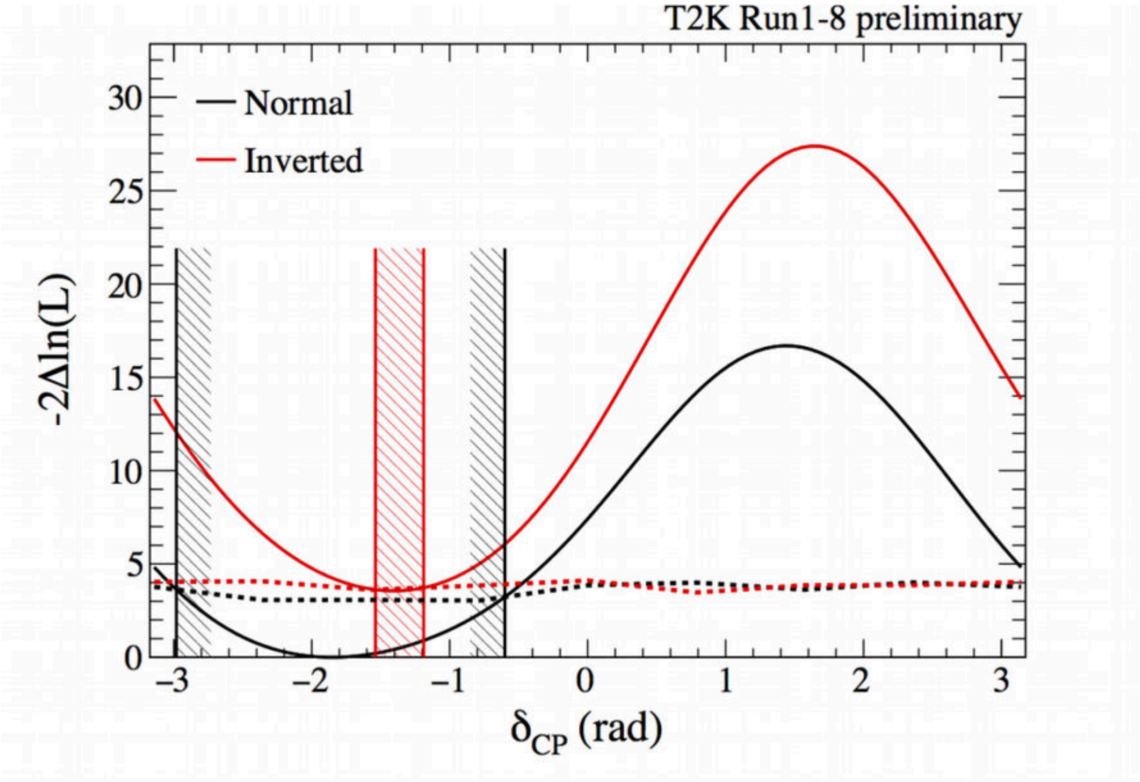


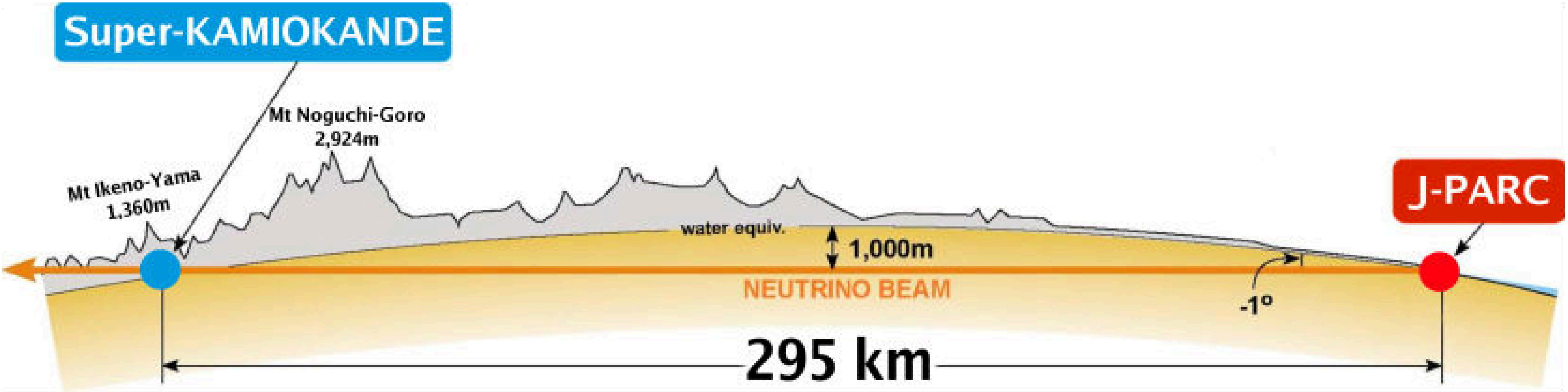
Simulating the behaviour of particles in a very complicated experimental setup:

Not something that can be done analytically!

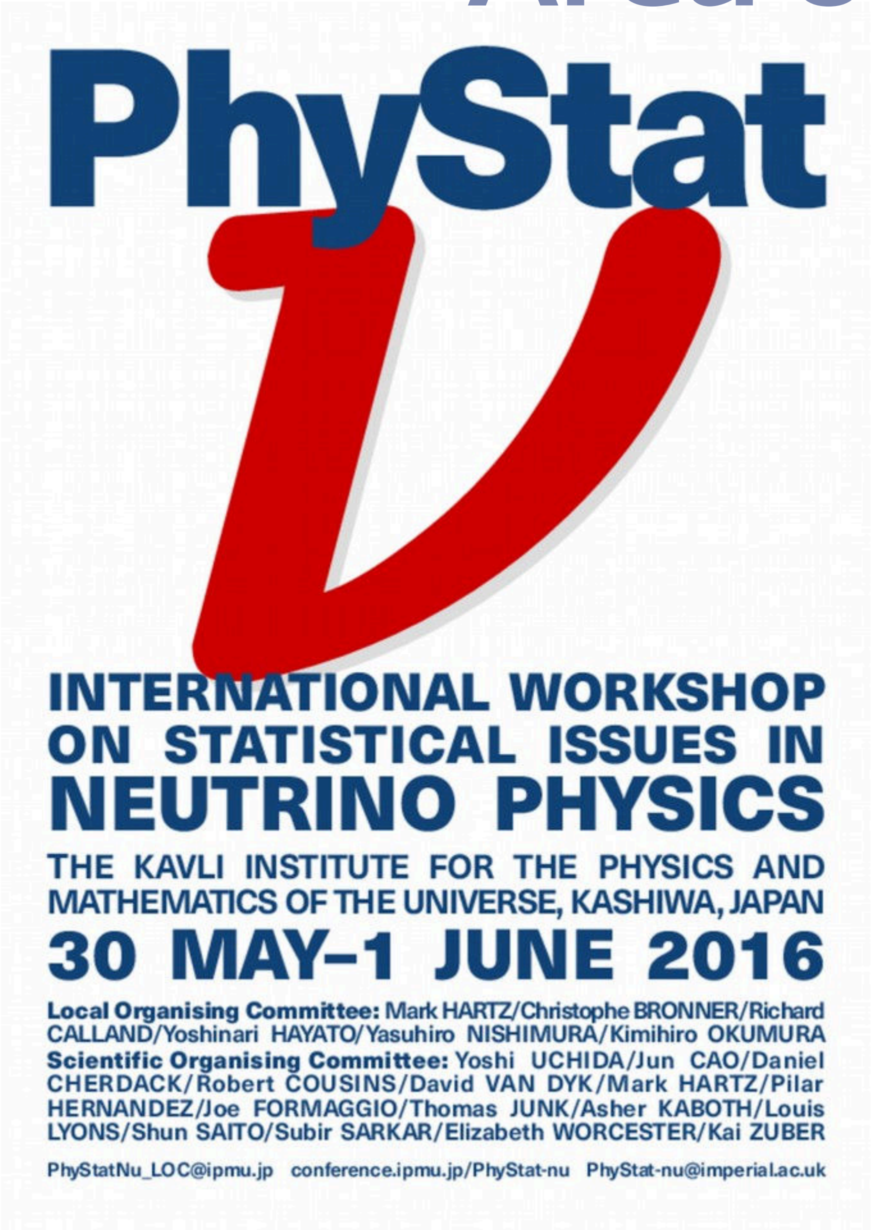
CP in Particle Physics

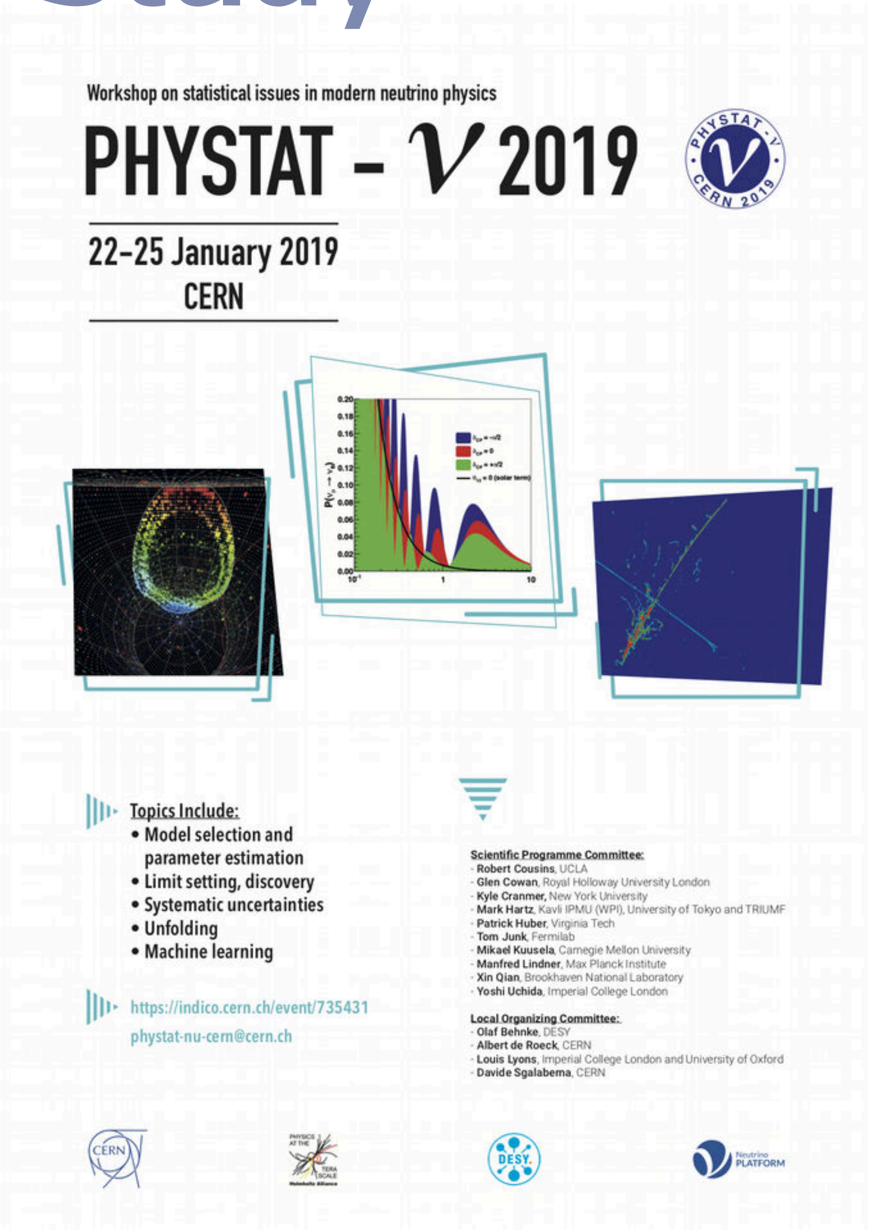
- Finding the statistically most likely values of “neutrino oscillation parameters” given data from multiple experiments





Statistical Methods: An Active Area of Study





Lecture 4

Today, we will discuss:

- Interpolation: what is it?
- Linear interpolation
- The alternative to interpolation
- Bilinear interpolation
- Lagrange polynomials
- The cubic spline
- Bonus topic: Unit Tests and Validation

Reminder

As with Pat

- Please ask questions‼
- Any time I'm speaking, cut in and ask!
- Any time I'm not, cut in and ask!
  - (well, wait for me to finish answering other peoples' questions before cutting in)

Interpolation

What is it?

- Given a set of data points (x_i, f_i), for $0 \leq i \leq n$, where $x_i < x_j$ for $i < j$, which represents some unknown function of x
- Interpolation allows one to obtain $f(x)$ for any value of x within the range that is covered by the set of x_i
- This is not a fully-determined problem
  - the information is simply not there to completely specify the values between the data points
  - therefore additional assumptions are needed
  - it is up to the physicist to decide what is adequate for the purposes
- "Interpolation" is defined as the case where **the result exactly passes through each of the (x_i, f_i) points that are given**
- also different to "approximation" of a function

Examples of Interpolation Algorithms

Linear Interpolation

- The simplest form of interpolation; just “connect the dots”
- Conceptually, that is all there is to it, but to code it up, you need to write down the expression for this

What are the pros and cons of this method?

- Simple and predictable
-
-

The Alternative to Interpolation

What else could we do (other than to go through every point (x_i, f_i))?

- "Function fitting"
  -
  -
  -

What are the fundamental differences between these two methods?

Examples of Interpolation Algorithms

Bi-Linear Interpolation

Interpolation is often needed for multi-dimensional functions

- For example, for $f(x, y)$
- One-dimensional linear interpolation can be extended into multiple dimensions
- For two dimensions, this is called bi-linear interpolation
- This is quite trivial; you can figure it out yourself!
- More in the lecture notes
- In general, it is good to think about how to extend one-dimensional methods into higher dimensions

Examples of Interpolation Algorithms

Lagrange Polynomials

Would be more useful to interpolate with a nice, smooth, function, not piecewise-straight lines

- How about a polynomial function that goes through all the points?
- **In general, for $n+1$ points (x_i, f_i), for $0 \leq i \leq n$, where $x_i \neq x_j$ for $i \neq j$, what is the minimum order of polynomial that can be made to pass through all the points?**
-
-
-

Examples of Interpolation Algorithms

Lagrange Polynomials

The Lagrange Polynomial:

$$P_n(x) = \sum_{i=0}^{n} \left(\prod_{j \neq i} \frac{x - x_j}{x_i - x_j} \right) f_i$$

Examples of Interpolation Algorithms

Lagrange Polynomials

So is the Lagrange polynomial the ultimate interpolation solution?

It's smooth and simple −what could be more straightforward than the lowest-order polynomial that connects all your points?

-
-
-
-

Examples of Interpolation Algorithms

Splines: General

As a general term, a “spline” is a function that matches polynomials along the data points, in piecewise fashion

You have as many degrees of freedom at your disposal as you want; but matching should be done so that the physics needs are met —resulting in a “stiffer” interpolant than the (not stiff at all) linear one

For example:

- a different quadratic function defined across each gap $x_i \leq x < x_{i+1}$ allows the first derivative to be matched at each point x_i
- but the second derivative will not be continuous (not enough degrees of freedom to allow them to be matched)
- a **cubic function** would allow us to match the second derivative at each x_i
- need more information than just the positions of two adjacent points

The Cubic Spline

- For the points x_i and x_{i+1}, $(0 \leq i \leq n-1)$
- Start by ensuring the polynomial passes through the two points
- Add corrections that are proportional to the second derivatives at each of the points x_i and x_{i+1}
- The second derivatives are not immediately known from the data points
- But can impose the continuity of the first derivative for on the above

The Cubic Spline

- This gives the following expression, with the second derivatives f_i'' as the only unknowns

$$\frac{x_i - x_{i-1}}{6} f_{i-1}'' + \frac{x_{i+1} - x_{i-1}}{3} f_i'' + \frac{x_{i+1} - x_i}{6} f_{i+1}'' = \frac{f_{i+1} - f_i}{x_{i+1} - x_i} - \frac{f_i - f_{i-1}}{x_i - x_{i-1}}$$

  where $i = 1 \,..\, n-1$
- $n - 1$ equations, $n + 1$ unknowns
- Need two boundary conditions: the f_0'' and f_n'' are obvious choices
- Setting both to 0 gives the “natural” spline
  - This is the form a thin massless and frictionless beam that is fixed through the points will take

The Cubic Spline

- Simultaneous equations for $f_0'' \ldots f_n''$
- How would one solve this? (We will ask you to do this in the Problem Sheet)
-
-
- Once you have these values, then the original equation can be used to find the value of $f(x)$ for any x with in the range
- The basic cubic spline works very well in general for any reasonable data set
- But it can still suffer from artefacts like “ringing”
- The onus is always on the physicist to check that any interpolation is doing what they want it to (no general algorithm will do that for you)
- Plot everything as you go is sound advice for all of the work you do!

Summary

Today we covered:

- Interpolation: connecting the dots the physicist's way
- Linear and bilinear interpolation
- Whether to fit or to interpolate (more later)
- Lagrange polynomials
- The Cubic Spline

A fuller mathematical treatment is given in the lecture notes (which you should be able to follow from what we discussed here)

Please remember that the problem sheets are intended to be an integral part of the course, not an optional extra! For this lecture we ask you to work through the fundamentals of Lagrange polynomial interpolation and cubic splines.