Interpolation of functions: spline interpolation

Nicola Seriani

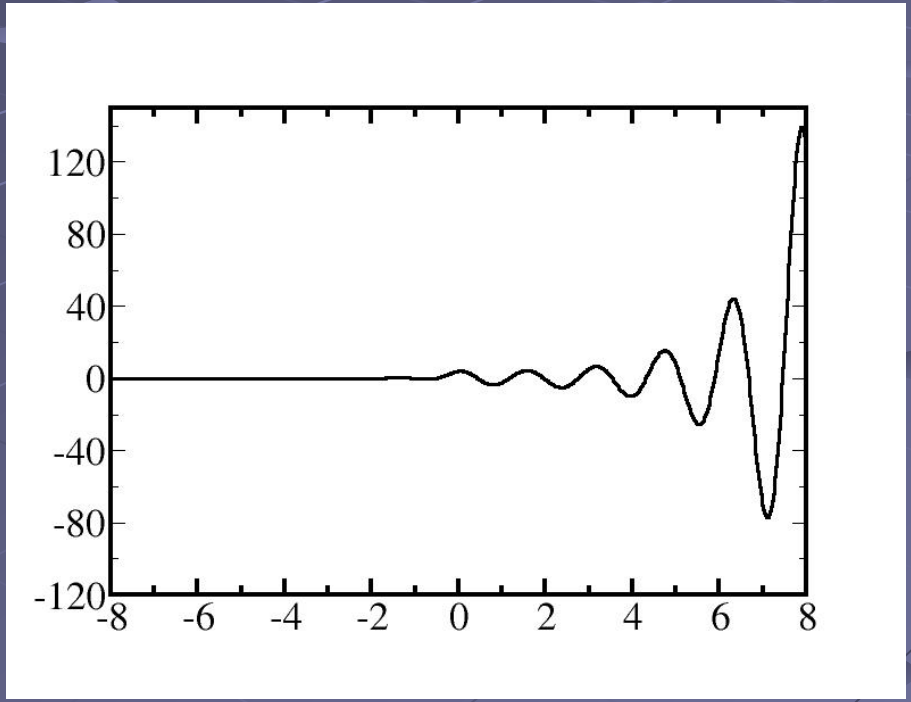
The Abdus Salam International Centre for Theoretical Physics,
Strada Costiera 11, 34151 Trieste, Italy

Need for interpolation of function

- Even in cases where the function is known, it may be computationally costly to calculate explicitly
- We might want to use a different, simpler method to calculate it
- Error must be small
- This is particularly true when we want to evaluate the function on a limited interval.

Need for interpolation of function

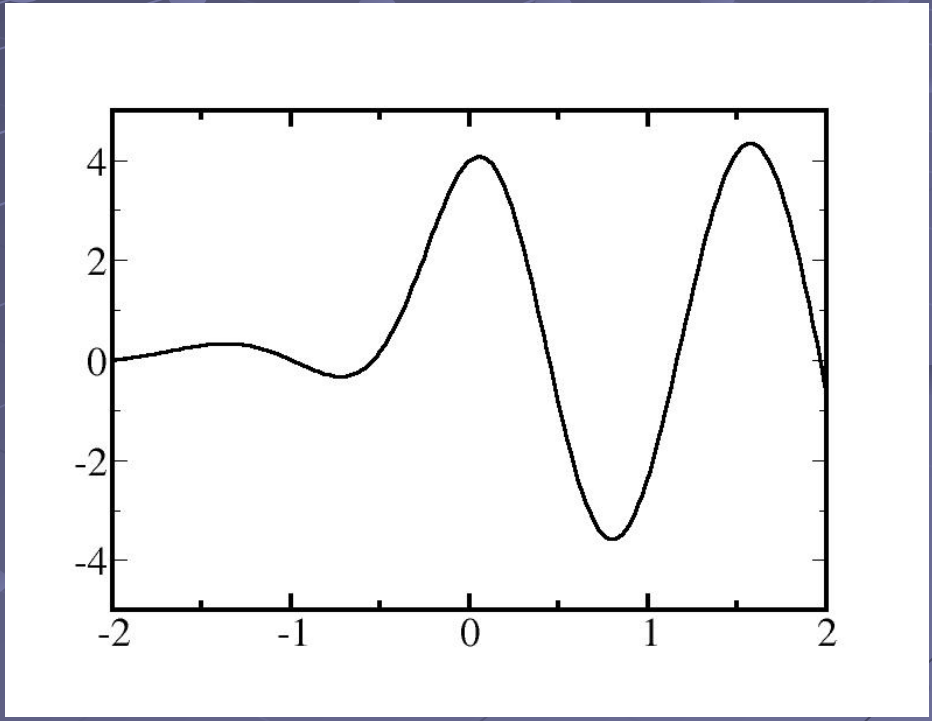
This is particularly true when we want to evaluate the function on a limited interval.



$f(x) = \exp(-x^2) + \ln(x+20)\cos(4x)\exp(x)/(1+x^2)$

Need for interpolation of function

This is particularly true when we want to evaluate the function on a limited interval.



$$f(x) = \exp(-x^2) + \ln(x+20)\cos(4x)\exp(x)/(1+x^2)$$

Interpolation with polynomials

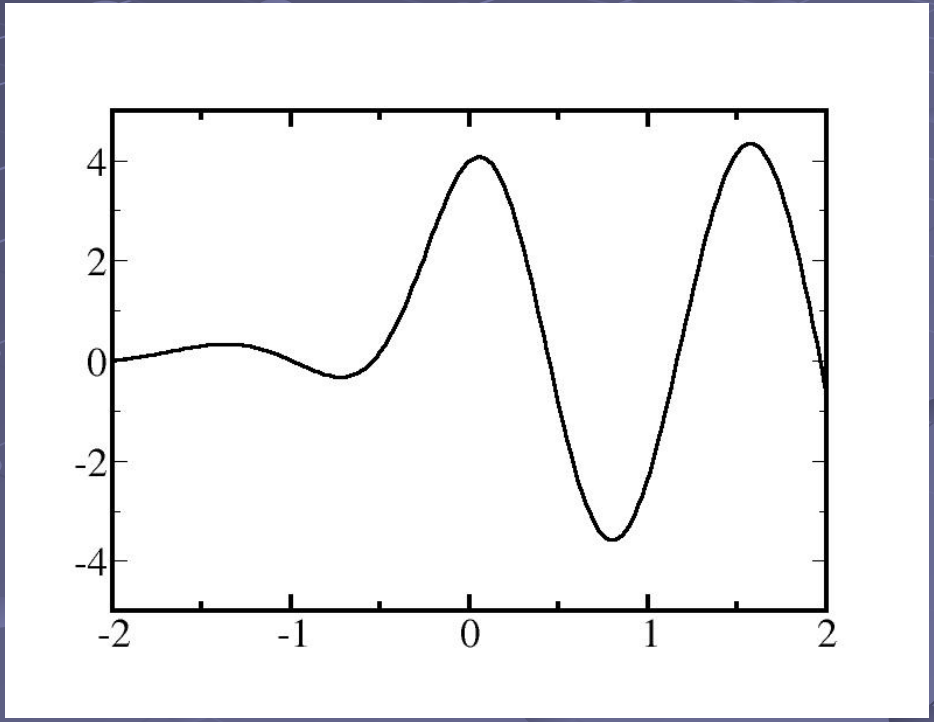
- Polynomials are relatively simple functions
- Flexible and general form
- Well behaved, differentiable,...

Interpolation with polynomials

- What is the polynomial that “best” interpolates a given function on an interval?
- Given functionl may be 1) an analytic expression or 2) expressed as values on a discrete number of points
- What does “best” mean? How do I calculate the polynomial? What is the error? When is it convenient?

Given function

- Analytic function may have a complex expression



$$f(x) = \exp(-x^2) + \ln(x+20)\cos(4x)\exp(x)/(1+x^2)$$

Given function

- Questions: if the function is known on a limited number of grid points (e.g. because it is stored in an array), how can I efficiently calculate it in points in-between?
- Need of accurate and efficient interpolation schemes....

x	f(x)
0	0
1	1
2	4
3	9
4	16

What is the value of this function in x = 1.5?

Polynomial interpolation

- For the moment, we are interested in finding a single polynomial that approximates the function on the whole interval

- Let's consider N (equispaced) points that span the interval, x_i (interpolation points), and the function at these points $y_i = f(x_i)$

Lagrange interpolation

- Let's start from polynomials $L_{n,j}(x)$ such that
- $L_{n,j}(x) = \delta_{ij}$
- They are called Lagrange polynomials. They are defined as
- $L_{n,j}(x) = \Pi_{k \neq j} (x-x_k)/(x_j-x_k)$
- Then the approximating polynomial is
- $p_n(x) = \Sigma_j f(x_j) L_{n,j}(x)$

James V. Lambers, Numerical Analysis

Lagrange interpolation

- $L_{n,j}(x) = \Pi_{k \neq j} (x-x_k)/(x_j-x_k)$
- Then the approximating polynomial is
- $p_n(x) = \Sigma_j f(x_j) L_{n,j}(x)$
- In case of two points: linear interpolation...

James V. Lambers, Numerical Analysis

Lagrange interpolation: some problems

- In Lagrange interpolation, the Lagrange polynomial needs to be re-evaluated if one point is added to the list.

- The degree of the polynomial increases with the number of points

Convergence and errors

- Theorem: if f(x) is differentiable, and $p_n(x)$ is its interpolating polynomial in the sense of Lagrange interpolation, then, for x in the interval,

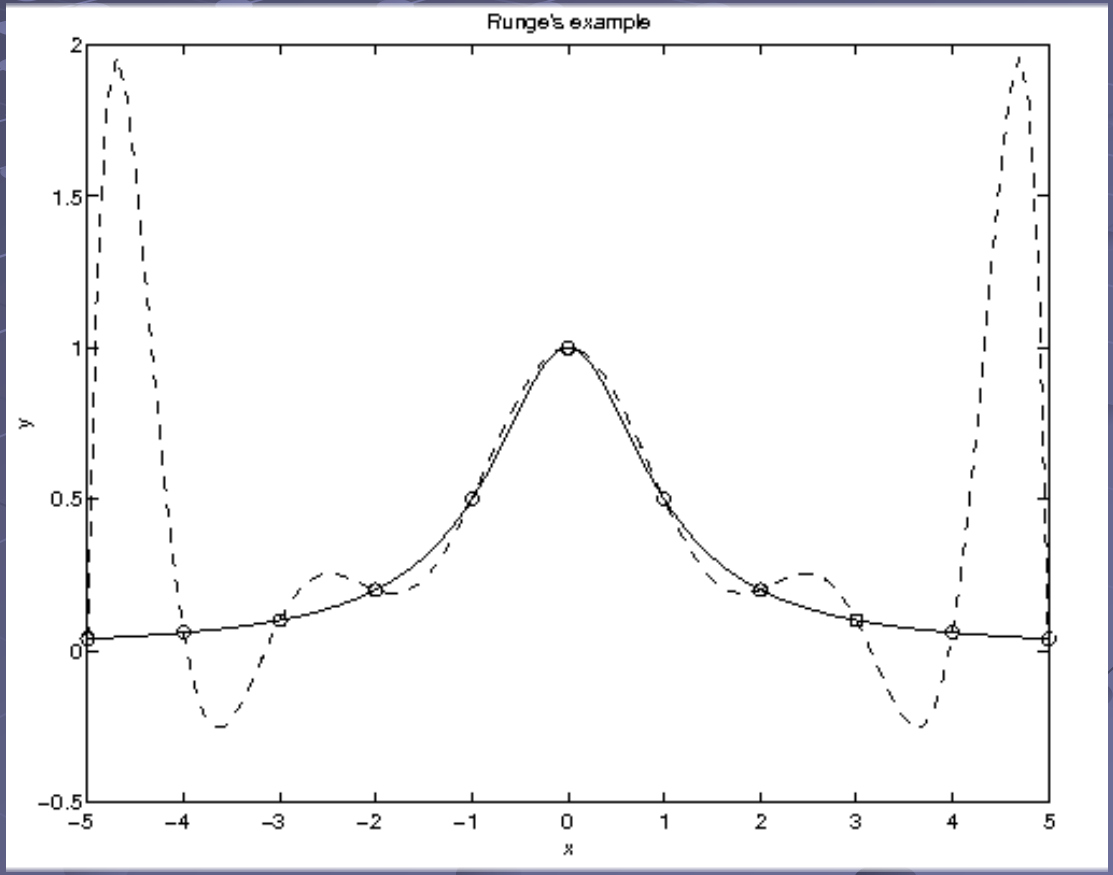
- $f(x) - p_n(x) = \Pi_j (x-x_j) f^{(n+1)}(\xi(x))/(n+1)!$

- (Demonstration)

- Problems for large $(x-x_j)$, large n (oscillations)

Runge's example:

- $f(x) = 1/(1+x^2)$ in the interval [-5, 5]; 11 points



J. V. Lambers, Numerical Analysis

Tenth-degree polynomial fails

Minimizing the errors: Chebyshev polynomials, Newton, Hermite,...

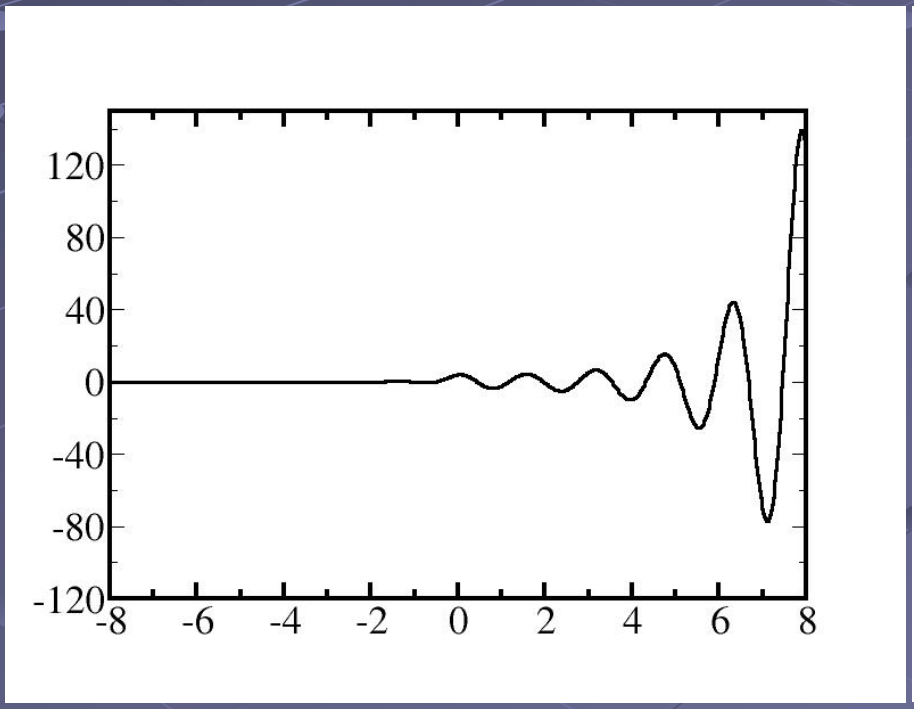
- $fT_k(x) = \cos(k\cos^{-1}(x))$...
- Chebyshev polynomials built to minimize the error in an interval. Useful if function is defined in that interval (or can be transformed into). For example, time evolution in quantum mechanics:
- $\Psi(r_1,r_2,r_3,\ldots,r_N,t) = e^{iH(r1,r2,r3,\ldots,rN)t}\,\Psi_0$

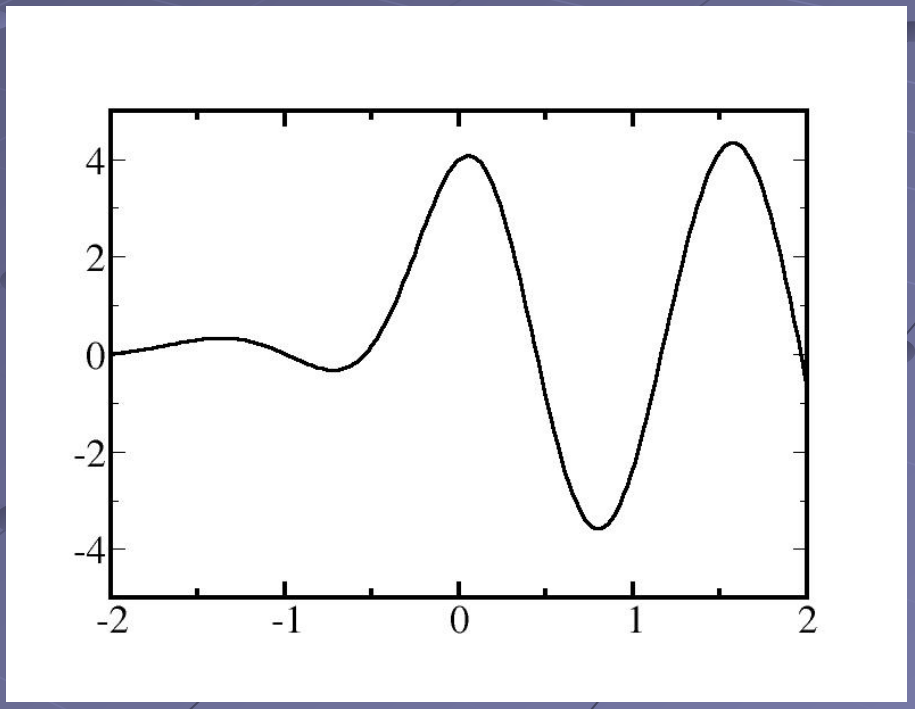
Splines

- We are going to follow a different path: use piece-wise polynomials (splines)

Interpolation

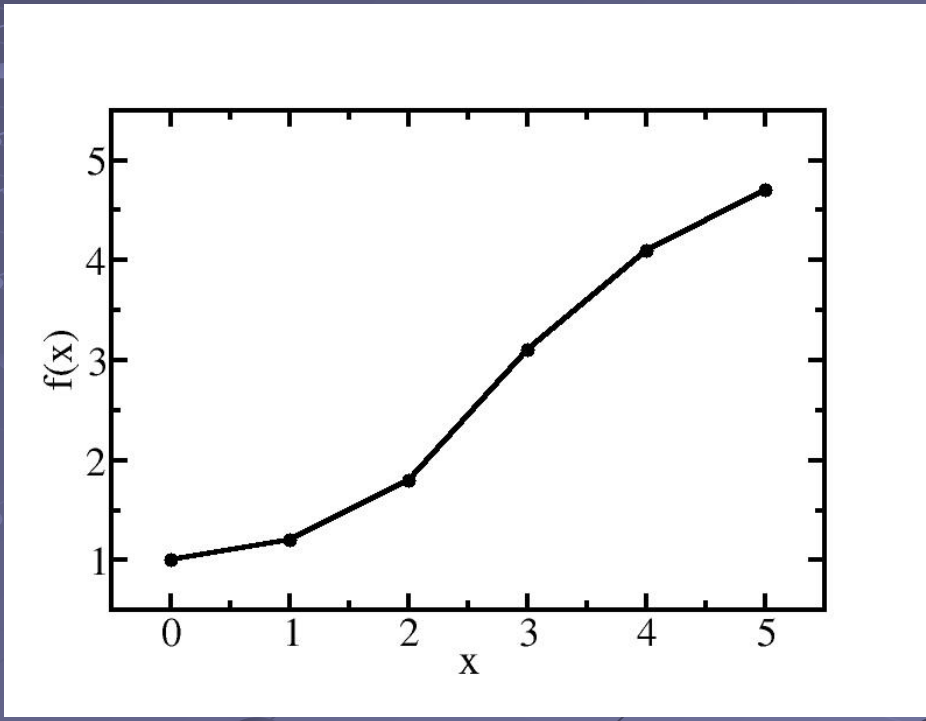
- Instead of aiming at a single polynomial describing the function over the whole interval (high degree, high error), the target is to use low-degree polynomials to describe the function in sub-intervals





Spline interpolation

x	f(x)
0	1
1	1.2
2	1.8
3	3.1
4	4.1
5	4.7



In each interval we have a different (low-degree) polynomial, they are smoothly linked together

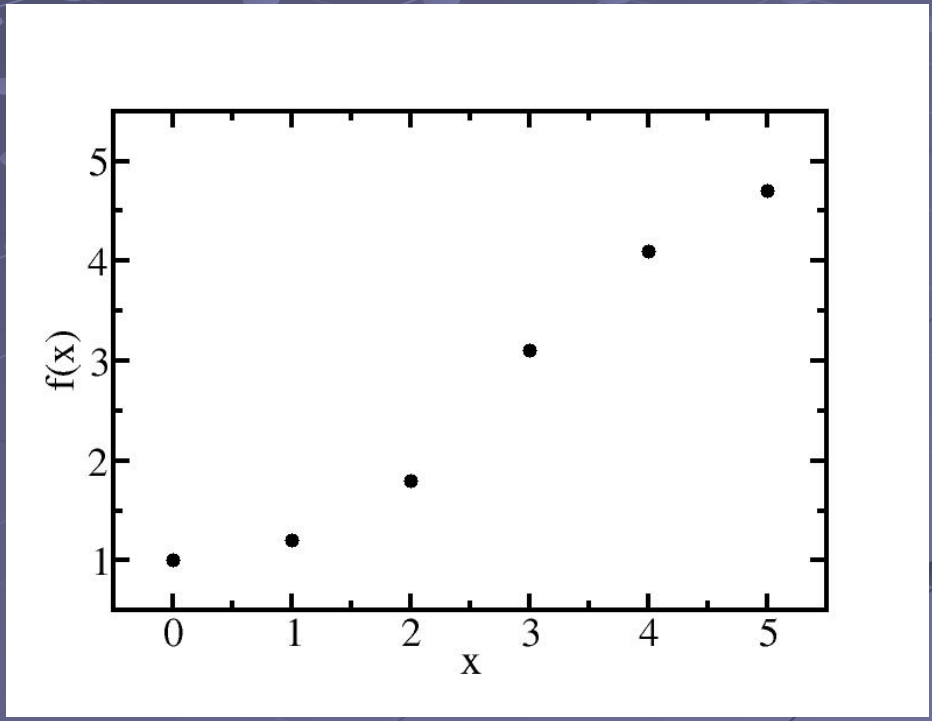
Linear interpolation

- Given values of the function on a finite set of points, I want to estimate values of the function at points in-between.

- Simplest approach: linear interpolation between consecutive points

Interpolation

- Given values of the function on a finite set of points, I want to calculate values of the function at points in-between.



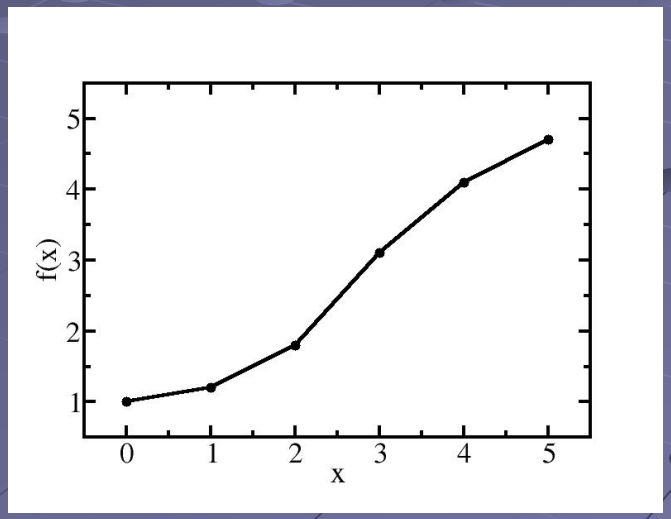
What is the value of this function in x = 1.5?

Linear interpolation

- We use a linear function between them. The linear function goes exactly through two consecutive points (x_i,y_i) and (x_{i+1},y_{i+1})
- $Y = A_i + B_i X$
- Imposing the two conditions

$y_i = A_i + B_i x_i$

$y_{i+1} = A_i + B_i x_{i+1}$

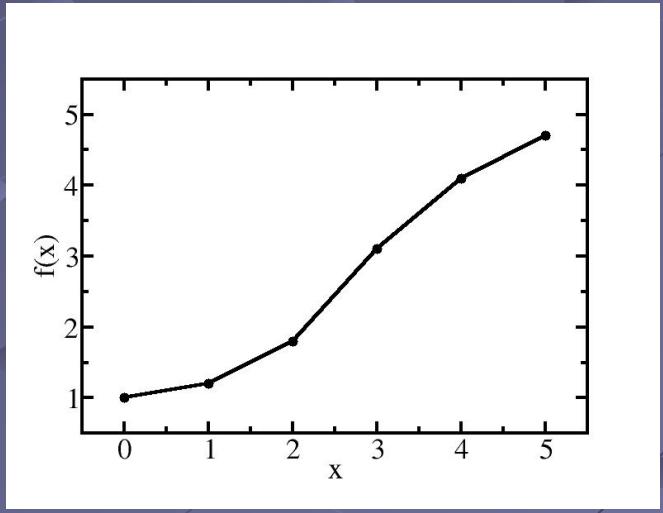


we obtain A_i and B_i, and the interpolating function:

$Y = y_i + (X-x_i)\,(y_{i+1} - y_i) / (x_{i+1} - x_i)$

Linear interpolation

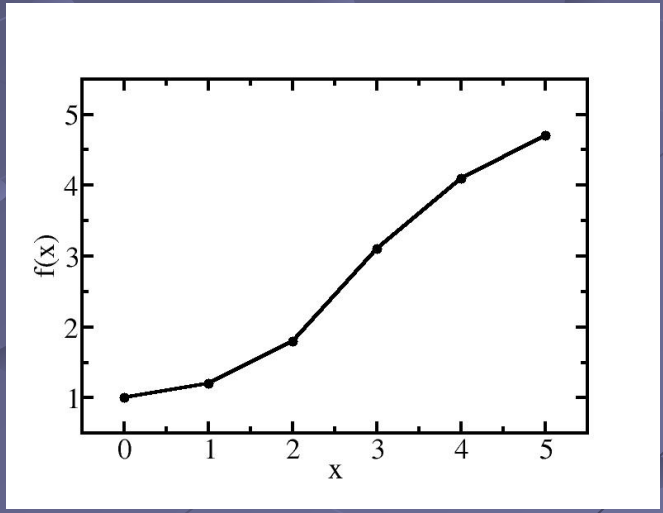
- We use a linear function between them. The linear function goes exactly through two consecutive points (x_i,y_i) and (x_{i+1},y_{i+1})
- $Y = y_i + (X-x_i)\,(y_{i+1} - y_i) / (x_{i+1} - x_i)$



Linear interpolation

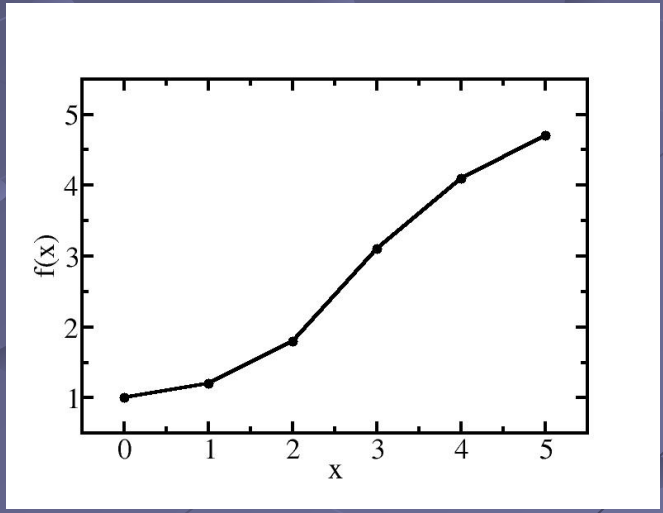
- $Y = y_i + (X-x_i)\,(y_{i+1} - y_i) / (x_{i+1} - x_i)$
- It may be re-written as
- $Y = y_i\,(x_{i+1} - X)/(x_{i+1} - x_i) + y_{i+1}\,(X-x_i) / (x_{i+1} - x_i)$

  $= A(X)\,y_i + B(X)\,y_{i+1}$



Linear interpolation

- $Y = y_i + (X-x_i)\,(y_{i+1} - y_i) / (x_{i+1} - x_i)$
- The interpolating function consists of a piece-wise linear function; it is continuous, but not differentiable at the grid points



Estimating errors in linear spline interpolation

- If the function has a continuous second derivative, from Rolle's theorem it follows that the error R satisfies:

$$|R| \leq (x_{i+1} - x_i)^2/8 \max_{xi \leq X \leq xi+1} |f''(X)|$$

Estimating errors in linear interpolation

- $|R| \leq (x_{i+1} - x_i)^2/8 \max_{xi \leq X \leq xi+1} |f''(X)|$

Demonstration:

Let's define $F(x) = f(x) - P(x) - g(x_A)(x - x_i)(x - x_{i+1})$

where g is defined such that $F(x_A) = 0$

This yields $g(x_A) = [f(x_A) - P(x_A)]/[(x_A - x_i)(x_A - x_{i+1})]$

Applying twice Rolle's theorem, one finds that there must be a point X such that $F''(X) = 0$, i.e.

$f''(X) = [f(x_A) - P(x_A)]/[(x_A - x_i)(x_A - x_{i+1})]$ 2

$[f(x_A) - P(x_A)] = \frac{1}{2} f''(X) [(x_A - x_i)(x_A - x_{i+1})]$

Estimating errors in linear interpolation

- $|R| \le (x_{i+1} - x_i)^2/8 \max_{xi \le X \le xi+1} |f''(X)|$

Demonstration:

$f(x_A) - P(x_A) = \frac{1}{2} f''(X) [(x_A - x_i)(x_A - x_{i+1})]$

$|f(x_A) - P(x_A)| = \frac{1}{2} |f''(X) [(x_A - x_i)(x_A - x_{i+1})]|$

$\le \frac{1}{2} |f''(X)| \frac{1}{4} (x_{i+1} - x_i)^2$

$\le 1/8 \max |f''(X)| (x_{i+1} - x_i)^2$

$|f(x_A) - P(x_A)| \le 1/8 \max|f''(X)| (x_{i+1} - x_i)^2$

Estimating errors in linear interpolation

- $|R| \leq (x_{i+1} - x_i)^2/8 \max_{xi \leq X \leq xi+1} |f''(X)|$

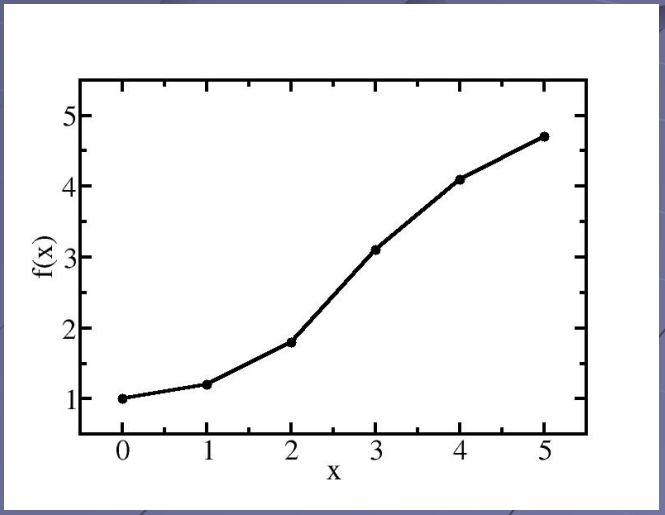
- Upper bound for the error depends on the function through f'', and on the grid of points through $(x_{i+1} - x_i)^2$

Linear interpolation

- We use a linear function between them.
- $Y = y_i + (X-x_i)\,(y_{i+1} - y_i) / (x_{i+1} - x_i)$
- $|R| \le (x_{i+1} - x_i)^2/8 \max_{xi \le X \le xi+1} |f''(X)|$
- Usually errors are not small:

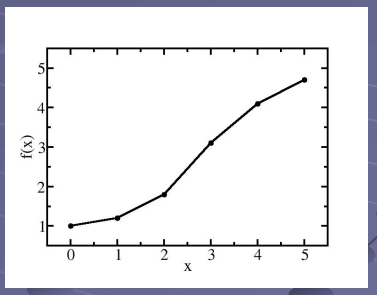
  we can do better: higher-order

  spline interpolation



Spline interpolation

- Linear interpolation gives an interpolating function that is continuous but generally not derivable in the table points: this might be not enough
- Moreover, smaller errors are often desirable
- Use higher-order polynomials instead: **spline interpolation**
- Linear interpolation is a special case of spline interpolation, with polynomials of degree 1



Spline interpolation

- In spline interpolation, one usually enforces continuity of derivatives
- To have continuous derivatives up to order N, a polynomial of degree ≥ N must be used
- If we call $f_i(x)$ the resulting interpolation function in the interval $[x_i, x_{i+1}]$, the continuity conditions then are

$$f_i(x_i) = f_{i+1}(x_i) = y_i$$

$$f_i'(x_i) = f_{i+1}'(x_i)$$

$$f_i''(x_i) = f_{i+1}''(x_i)$$

...

Spline interpolation

- Continuity conditions are enforced

  $f_i(x_i) = f_{i+1}(x_i)$

  $f_i'(x_i) = f_{i+1}'(x_i)$

  $f_i''(x_i) = f_{i+1}''(x_i)$

- An usual choice is to use polynomials of order 3 (**cubic splines**)
- $f_i(x) = a_i + b_i x + c_i x^2 + d_i x^3$
- (4 parameters for each interval)
- Also because often physical laws expressed in terms of first and second derivatives F = m r''

Conditions on parameters for cubic splines

- The conditions on the parameters are

  $f_i(x_i) = y_i$

  $f_i(x_{i+1}) = y_{i+1}$

  $f_i'(x_i) = f_{i+1}'(x_i)$

  $f_i''(x_i) = f_{i+1}''(x_i)$

- If there are N intervals (i goes from 0 to N), this gives N+N+(N-1)+(N-1) = 4N-2 conditions, but there are 4N parameters: conditions at the boundaries x_0 and x_N

Conditions on parameters for cubic splines

- If there are N intervals (i goes from 0 to N), this gives N+N+(N-1)+(N-1) = 4N-2 conditions, but there are 4N parameters: conditions at the boundaries x_0 and x_N

① Natural spline: the highest derivatives are set to zero in these points

② Clamped spline: the first derivatives in these points are given

Conditions on parameters for cubic splines

① Natural spline: the highest derivatives are set to zero in these points

$f''(x_0) = 0$

$f''(x_N) = 0$

Conditions on parameters for cubic splines

① Natural spline: the highest derivatives are set to zero in these points

② Clamped spline: the first derivatives in these points are given

$f'(x_0) = A$

$f'(x_N) = B$

Cubic splines

- $f_i(x) = a_i + b_i (x-x_i) + c_i (x-x_i)^2 + d_i (x-x_i)^3$
- 4 parameters for each interval
- From the conditions shown before, an algorithm to determine the parameters can be developed
- There are many approaches, also depending on boundary conditions

Cubic splines: finding the coefficients

- Here one derivation for natural boundary conditions (second derivatives are 0 in x_0 and x_N)

  (from H.F. Walker, Worcester Polytechnic Institute)

- $f_i(x) = a_i + b_i (x-x_i) + c_i (x-x_i)^2 + d_i (x-x_i)^3$
- Conditions are $f_i(x_i) = y_i$ and $f_i(x_{i+1}) = y_{i+1}$
- This gives $a_i = y_i$; then
- $y_i + b_i (x_{i+1}-x_i) + c_i (x_{i+1}-x_i)^2 + d_i (x_{i+1}-x_i)^3 = y_{i+1}$

Cubic splines: finding the coefficients

- Then we use the conditions on the derivatives
- $f'_i(x) = b_i + 2c_i (x-x_i) + 3d_i (x-x_i)^2$
- $f''_i(x) = 2c_i + 6d_i (x-x_i)$
- First derivative: $b_i + 2c_i (x_{i+1}-x_i) + 3d_i (x_{i+1}-x_i)^2 = b_{i+1}$
- Second derivative: $2c_i + 6d_i (x_{i+1}-x_i) = 2c_{i+1}$

Cubic splines: finding the coefficients

- Summarizing:

1. Function: $y_i + b_i (x_{i+1}-x_i) + c_i (x_{i+1}-x_i)^2 + d_i (x_{i+1}-x_i)^3 = y_{i+1}$
2. First derivative: $b_i + 2c_i (x_{i+1}-x_i) + 3d_i (x_{i+1}-x_i)^2 = b_{i+1}$
3. Second derivative: $2c_i + 6d_i (x_{i+1}-x_i) = 2c_{i+1}$

- 1., 2. and 3. are a set of linear equations that fully determine the coefficients
- 4N equations for 4N coefficient: slow if N is large

An alternative approach: B-splines

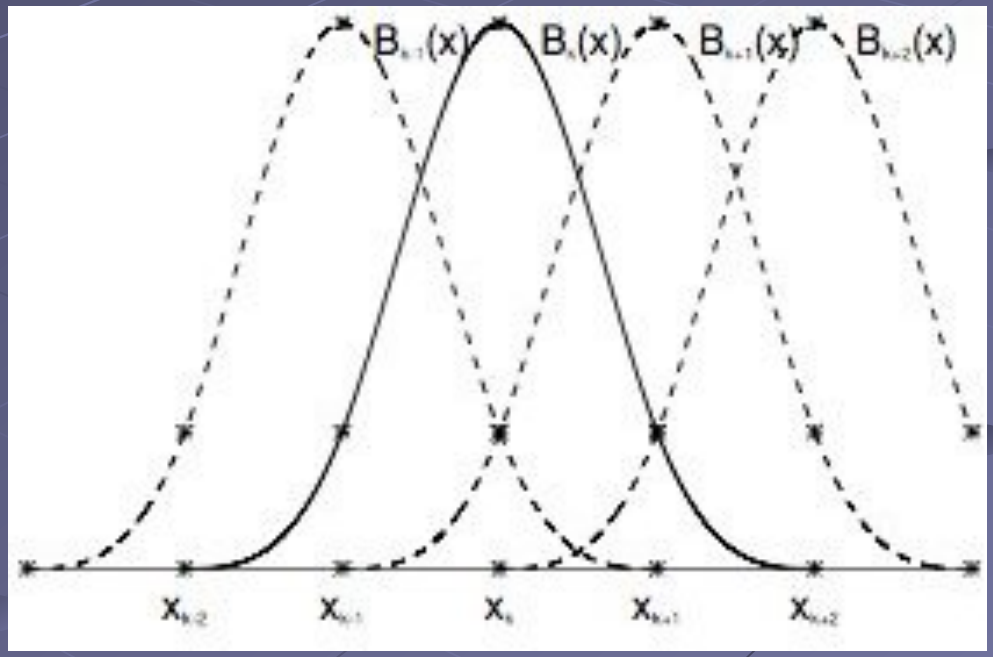
- A more systematic approach to splines:
- We construct our piecewise polynomial as a linear combination of polynomials with the right properties (the basis polynomials -> basis spline)
- Our range is $[x_0,x_N]$
- $h = (x_N-x_0)/N$

B-splines

- Our range is $[x_0,x_N]$
- $h = (x_N-x_0)/N$
- Consider the polynomial B(x) such that
- $B(x) = 0$ for $x<-2h$
- $B(x) = 1/6(2h+x)^3$ for $-2h<x<-h$
- $B(x) = 2/3h^3 – ½ x^2 (2h+x))$ for $-h<x<0$
- $B(x) = 2/3h^3 – ½ x^2 (2h-x))$ for $0<x<h$
- $B(x) = 1/6(2h-x)^3$ for $h<x<2h$
- $B(x) = 0$ for $x>2h$

B-splines

- B(x) is non-zero over 4 intervals, it is continuous, etc.
- Let's define one such polynomial for each x_i point of our grid $B_i(x) = B(x) – (x – kh – x_0)$



from Sue Liu, University of Edinburgh

B-splines

- We can now express our polynomial as a linear combination of basis polynomials:
- $f(x) = \Sigma_k a_k B_k(x)$
- The continuity conditions are automatically satisfied
- The other conditions are that f(x) must be equal to our function in the grid points: $f(x_i) = y_i$
- This gives set of linear equations for the a_k's:
- $a_{i-1} + 4 a_i + a_{i+1} = 6/h^3 y_i$

B-splines

- This gives set of linear equations for the a_k's:
-
- $a_{i-1} + 4\,a_i + a_{i+1} = 6/h^3\, y_i$
- N+1 equations for N+1 coefficients (we had 4N coefficients)