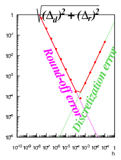
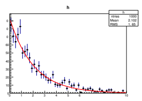
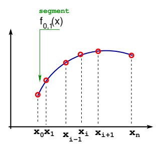
Computational Physics

numerical methods with C++ (and UNIX)

Fernando Barao

Instituto Superior Tecnico, Dep. Fisica

email: barao at lip.pt

Numerical methods

- ✔ System of linear equations
  - ► Gauss elimination
  - ► LU decomposition
  - ► Gauss-Seidel method
- ✔ Interpolation
  - ► Lagrange interpolation
  - ► Newton method
  - ► Neville method
  - ► Cubic spline
- ✔ Numerical derivatives
  - ► First derivative $\mathcal{O}(h^2)$, $\mathcal{O}(h^4)$
  - ► Second derivative $\mathcal{O}(h^2)$, $\mathcal{O}(h^4)$
  - ► Derivative by interpolation
- ✔ Numerical integration
  - ► Newton-Cotes: trapezoidal and Simpson rules
  - ► Gaussian quadrature
- ✔ Monte-Carlo methods

Numerical methods

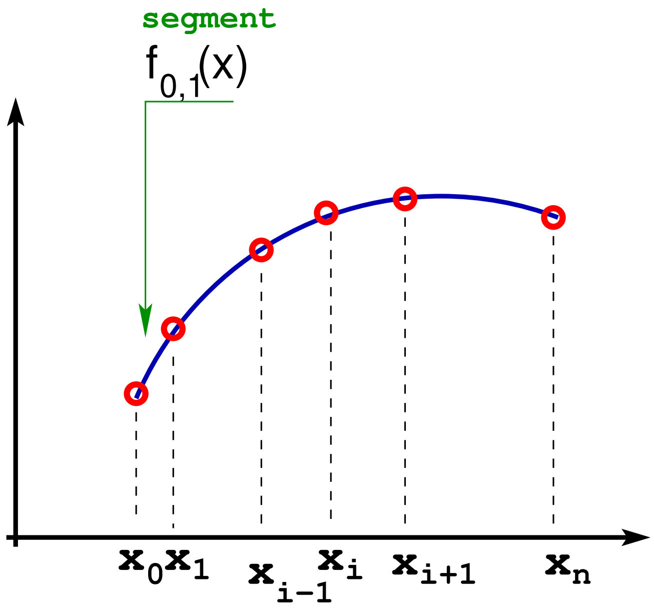
- System of linear equations
  - Gauss elimination
  - LU decomposition
  - Gauss-Seidel method
- Interpolation
  - Lagrange interpolation
  - Newton method
  - Neville method
  - Cubic spline
- Numerical derivatives
  - First derivative $\mathcal{O}(h^2)$, $\mathcal{O}(h^4)$
  - Second derivative $\mathcal{O}(h^2)$, $\mathcal{O}(h^4)$
  - Derivative by interpolation
- Numerical integration
  - Newton-Cotes: trapezoidal and Simpson rules
  - Gaussian quadrature
- Monte-Carlo methods

Limitations of polynomial interpolation

- The need of knowing with a better precision an interpolation carries the solution of adding more and more points to our interpolation
  - ☞ a polynomial interpolation passing through a large number of points (degree higher than $\sim 5, 6$) can give a wrong interpolation in some segments due to *wild* oscillations
  - ☞ if the number of points (knots) is large, an eventual linear interpolation by segments is enough!
  - ☞ otherwise a degree 3 to 6 polynomial interpolation by segment
- polynomial extrapolation (interpolating outside the range of data points) is dangerous!

Cubic spline method

- The interpolation can be performed in a given segment $[x_i, x_{i+1}]$ using a **cubic polynomial** (4 parameters to find)
- Apart from the two points data associated to the segment we ask for continuity of the 1st and 2nd derivatives at the knot x_{i+1}, i.e., the intersection of two segments
  - ☞ no bending at the end points (x_0 and x_n) $\Rightarrow$ 2nd derivative=0



- **The spline will be a piecewise cubic curve, put together from the n cubic polynomials:** $f_{0,1}(x), f_{1,2}(x), \cdots, f_{n-1,n}(x)$

Cubic spline method (cont.)

- the continuity of the 2nd derivative of the spline at knot i gives: $f^{j\prime\prime}_{i-1,i}(x_i) = f''_{i,i+1}(x_i) = K_i \; (K_0 = K_n = 0)$
- starting from the second derivative expression that is a linear polynomial, we can compute the coefficients of $f_{i,i+1}(x)$,

$$f''_{i,i+1}(x) = \frac{K_i(x - x_{i+1}) - K_{i+1}(x - x_i)}{x_i - x_{i+1}}$$

Integrating now twice:

$$f'_{i,i+1}(x) = \frac{1}{x_i - x_{i+1}}\left[\frac{K_i}{2}(x - x_{i+1})^2 - \frac{K_{i+1}}{2}(x - x_i)^2\right] + A$$
$$f_{i,i+1}(x) = \frac{1}{x_i - x_{i+1}}\left[\frac{K_i}{6}(x - x_{i+1})^3 - \frac{K_{i+1}}{6}(x - x_i)^3\right] + Ax + B$$

Redefining the constants:

$$f_{i,i+1}(x) = \frac{1}{x_i - x_{i+1}}\left[\frac{K_i}{6}(x - x_{i+1})^3 - \frac{K_{i+1}}{6}(x - x_i)^3\right] + A(x - x_{i+1}) + B(x - x_i)$$

Cubic spline method (cont.)

✔ The extreme values of the function on the segment provide A and B:

$$f_{i,i+1}(x_i) = y_i \quad \Rightarrow \quad \frac{1}{x_i - x_{i+1}}\left[\frac{K_i}{6}(x_i - x_{i+1})^3\right] + A(x_i - x_{i+1}) = y_i$$

$$\Rightarrow \quad A = \frac{y_i}{x_i - x_{i+1}} - \frac{K_i}{6}(x_i - x_{i+1})$$

$$f_{i,i+1}(x_{i+1}) = y_{i+1} \quad \Rightarrow \quad \frac{1}{x_i - x_{i+1}}\left[-\frac{K_{i+1}}{6}(x_{i+1} - x_i)^3\right] + B(x_{i+1} - x_i) = y_{i+1}$$

$$\Rightarrow \quad B = \frac{y_{i+1}}{x_i - x_{i+1}} - \frac{K_{i+1}}{6}(x_i - x_{i+1})$$

✔ Replacing A and B in the segment expression before, it becomes:

$$\begin{aligned} f_{i,i+1}(x) = & \frac{K_i}{6}\left[\frac{(x-x_{i+1})^3}{x_i - x_{i+1}} - (x - x_{i+1})(x_i - x_{i+1})\right] \\ & - \frac{K_{i+1}}{6}\left[\frac{(x-x_i)^3}{x_i - x_{i+1}} - (x - x_i)(x_i - x_{i+1})\right] + \frac{y_i(x-x_{i+1}) - y_{i+1}(x-x_i)}{x_i - x_{i+1}} \end{aligned}$$

✔ The second derivatives K_i values of the spline in the interior knots, are obtained from the first derivative condition:

$$f'_{i-1,i}(x_i) = f'_{i,i+1}(x_i)$$

Cubic spline: Problem

Utilizar o método do "cubic spline" para determinar o valor de $y(1.5)$, dados os seguintes valores:

x	1	2	3	4	5
y	0	1	0	1	0

Numerical methods

- ✔ System of linear equations
  - ► Gauss elimination
  - ► LU decomposition
  - ► Gauss-Seidel method
- ✔ Interpolation
  - ► Lagrange interpolation
  - ► Newton method
  - ► Neville method
  - ► Cubic spline
- ✔ Numerical derivatives
  - ► First derivative $\mathcal{O}(h^2)$, $\mathcal{O}(h^4)$
  - ► Second derivative $\mathcal{O}(h^2)$, $\mathcal{O}(h^4)$
  - ► Derivative by interpolation
- ✔ Numerical integration
  - ► Newton-Cotes: trapezoidal and Simpson rules
  - ► Gaussian quadrature
- ✔ Monte-Carlo methods

Numerical methods

- ✔ System of linear equations
  - ► Gauss elimination
  - ► LU decomposition
  - ► Gauss-Seidel method
- ✔ Interpolation
  - ► Lagrange interpolation
  - ► Newton method
  - ► Neville method
  - ► Cubic spline
- ✔ Numerical derivatives
  - ► First derivative $\mathcal{O}(h^2)$, $\mathcal{O}(h^4)$
  - ► Second derivative $\mathcal{O}(h^2)$, $\mathcal{O}(h^4)$
  - ► Derivative by interpolation
- ✔ Numerical integration
  - ► Newton-Cotes: trapezoidal and Simpson rules
  - ► Gaussian quadrature
- ✔ Monte-Carlo methods