Linear spline interpolation

Nicola Seriani

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

Linear spline interpolation

- We are now going to use linear interpolation for the exponential exp(x) for x in [-10,10]
- This time we are going to be more quantitative: to this aim, most of the code is provided in the Input directory
- Two main files: tester.c and spline.c

Linear spline interpolation

- In src/tester.c you find the main code. It generates a number num of random values between x_0 and x_{fin}, calculates the exponential with the built-in function and with the subroutine you are going to write, prints average error and timing information
- The coefficients for the spline interpolation are generated in the subroutine genspline
- Task: write the subroutine that uses linear spline interpolation to calculate exp(x) (see spline.c)

Linear spline interpolation

- Task: write the subroutine that uses linear spline interpolation to calculate exp(x) (see spline.c)
- Sub-task 1: understand how the piece-wise linear function is generated in genspline
- Sub-task 2: write a subroutine that, given x, calculates exp(x); given x, it must understand in which interval 'i' (xar[i-1], xar[i]) it falls and use the appropriate linear function f (x) = a[i] +b[i] (x-xar[i-1]). The appropriate position for this is shown by

  /* WRITE YOUR SUBROUTINE HERE */

Technical notes

- Compile with ‘make 32bit-fpu-gcc’. Alternatively, compile with ‘make 64bit-fpu-gcc-modern’ **
- Run with ‘./Obj_32bit-fpu-gcc/tester num repnum’, where num is the number of points x for which exp(x) is calculated, repnum is the number of repetitions
- A good choice is
  ‘./Obj_32bit-fpu-gcc/tester 10000 1’
- spl_exp is our approximation, exp is the built-in function

** More on compiling options:
https://gcc.gnu.org/onlinedocs/gcc/x86-Options.html

Linear spline interpolation

- Task: we would like to understand how the performance and accuracy of our interpolation depend on the parameters used; increase the nmax number of intervals in which the [-10,10] is divided, from 100 to 2000: how is the error changing? How is the execution time changing?

- Is linear interpolation worth the effort?