Programing Techniques for Scientific Simulations: Report B

Marion Baumgartner

November 27, 2012

1 Simpson Integration

Simpson's rule is used to integrate several different function such as

$$\begin{aligned} f(x) &= 1 \\ f(x) &= x \\ f(x) &= x^2 \\ f(x) &= \sin(x) \\ f(x) &= \sin(5 \cdot x) \end{aligned}$$

The functions are all integrated in an interval $[0, \pi]$ using $N = 100$ steps. The basic formula used is

$$Q(f) = \frac{h}{3} \cdot \left(\frac{1}{2} f(x_0) + \sum_{k=1}^{N-1} f(x_k) + 2 \sum_{k=1}^{N} f\left(\frac{x_{k-1} + x_k}{2} \right) + \frac{1}{2} f(x_N) \right) \tag{1}$$

$$h = \frac{b-a}{N} \qquad\qquad x_k = a + k \cdot h$$

Where $[a, b]$ is the interval to be integrated over and N determines the number of bins (N has to be even). Further we have that

$$\int_a^b f(x)dx = Q(f). \tag{2}$$

I used several different methods to program the algorithm of the Simpson function. These were

1. Hard-Core function
2. function pointers
3. (template) function objects
4. virtual functions

f(x)	1	x	x^2	$\sin(x)$	$\sin(5 \cdot x)$
Hard-Core function	4	5	7.15	9	28
function pointers	9.1	12.9	14	44	45
function objects	10	5.69	6.16	28	22
virtual functions	35	39	42.2	28.8	30

Table 1: Run times of the programs. All the times are given in 10^{-6}sec.

The Run times of the different methods with the different functions are shown in table 1 and they are graphically presented in figure 1.

As we can see from the table for the functions given the fastest is the hard-cored function method. The slowest program for simple function such as $f(x) = 1$; $f(x) = x$; and $f(x) = x^2$ is the one done with virtual functions. For more complex functions ($f(x) = \sin(x)$ and $f(x) = \sin 5(x)$) the program using function pointer is the slowest. the reason why the program using virtual functions is slow is because the polymorphic function resolution occurs at runtime and indirect virtual functions tables calls are made only at runtime. This causes time to be similar for all functions $f(x)$. An Improvement in the program using Hard-Core function and function pointers could perhaps e achieved by inlining the function call of $f(x)$

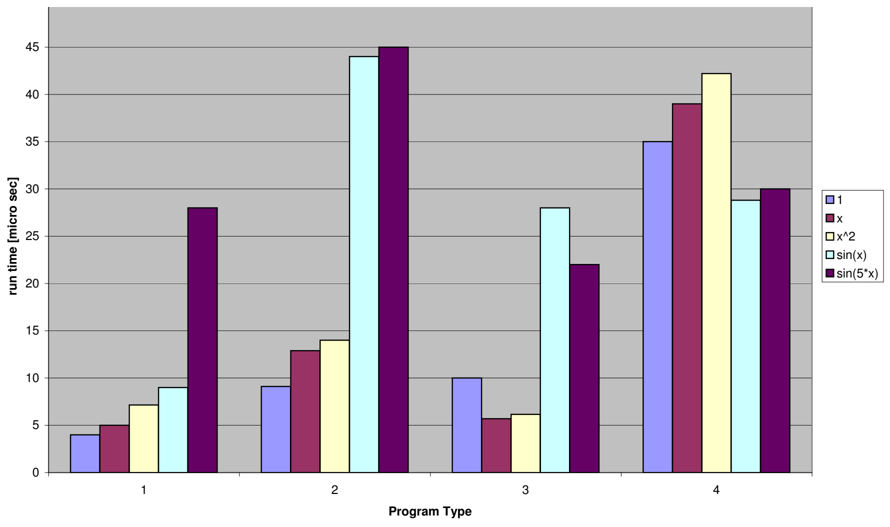

Figure 1: run time of the different implementations of the Simpson method. Method 1 is the Hard-Core function method; 2 is the version using function pointers; 3 uses (template) function objects and 4 is programed with virtual functions.