# Project 1, SF2565

Hanna Hultin, TTMAM2
Mikael Persson, TTMAM2

September 27, 2016

**Task 1**

Consider the $2N$ degree Taylor polynomial for $\cos x$

$$\cos x \approx p(x) = \sum_{n=0}^{N} (-1)^n \frac{x^{2n}}{(2n)!}$$

$$= 1 + (-1)\frac{x^2}{2!} + (-1)^2\frac{x^4}{4!} + \cdots + (-1)^N \frac{x^{2N}}{(2N)!}$$

$$= 1 - \frac{x \cdot x}{2 \cdot 1}\left(1 - \frac{x \cdot x}{4 \cdot 3}\left(1 - \ldots \left(1 - \frac{x \cdot x}{(2N)(2N-1)}\right)\ldots\right)\right).$$

Hence, the polynomial may be evaluated backwords using the following scheme

$$b_N = 1 - \frac{x \cdot x}{2N(2N-1)}$$

$$b_n = 1 - \frac{x \cdot x}{2n(2n-1)}b_{n+1}, \quad n = N-1, N-2, \ldots, 2, 1$$

$$b_1 = p(x).$$

This is Horners' algorithm adjusted for the fact that each second term in the polynomial vanishes. Similarly for $\sin x$ the polynomial may be computed up to degree $2N + 1$ by

$$b_N = 1 - \frac{x \cdot x}{2N(2N+1)}$$

$$b_n = 1 - \frac{x \cdot x}{2n(2n+1)}b_{n+1}, \quad n = N-1, N-2, \ldots, 2, 1$$

$$b_1 = x \cdot p(x).$$

**Task 2**

In this task we implement Adaptive Simpson Integration to integrate the function $f(x) = 1 + \sin e^{3x}$ between $a = -1$ and $b = 1$. We use the functions iSimpson and i2Simpson to compute the approximate integrals named $I$ and $I_2$ in the project description.

We implemented the Adaptive Simpson Integration both by using recursion, in the function iRecursive, and without recursion, in the function asi.

In asi we look at it as if we have a tree instead of recursion. We do not implement an actual tree structure, instead we just keep track of which node we are currently in. When we go to the left child of a node we change $b$ to $(a + b)/2$ and when we go to

the right child we change $a$ to $(a+b)/2$ and we also half the tolerance for every child. We can also go up from a child to its parent by doing the reverse computations.

We start in the root node with number 1 and then we go to the left child by using $leftnode = 2 * node$ until we get that errest $< 15 * tol$, and then we found our first leaf. When we find a leaf we keep going up in the tree by using $parentnode = floor(0.5 * node)$ until we find the first even node, there we go up one more step and then to the right with $rightnode = 2 * node + 1$. From there we start going to the left again until we find another leaf. If we can not find an even node before the root node when we go up from a leaf, we have been through the whole tree and can stop. We get the approximate value of the integral by summing $I_2$ for each leaf of the tree.

In Figure 1 we can see how we go through the tree in the case when $tol = 0.01$. We start from the root and go down to the left until we reach the first leaf which is node 4. Then we go up to 2 and down to 5 again which also is a leaf. Then we go up to 2, up to 1, down to 3, down to 6 and we found another leaf. Then up to 3 again, down to 7, down to 14 and we found another leaf. Then we go up to 7 again, down to 15, down to 30, down to 60 and we found another leaf. So we go up to 30 and down to the next leaf 61. Then up to 30, up to 15, down to 31 and down to the next leaf 62. Back to 31 and down to our last leaf 63. Then we go up to 31, up to 15, up to 7, up to 3, up to 1 and since we have not found a single even node (63,31,15,7,3,1 all uneven) we have gone through our whole tree and are done.
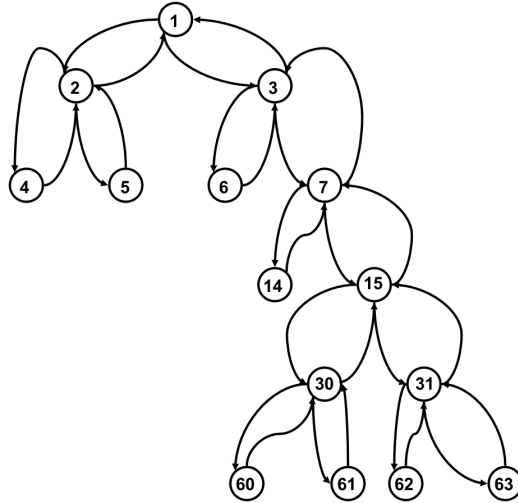


Figure 1: Illustration of how we go through the nodes for $tol = 0.01$ to do Adaptive Simpson Integration without recursion.

```
for (int i=0; i<iterations; i++)
{
  do something
}
```