# Example: Minimum order Taylor polynomial to achieve a given precision

Let  $f(x) = e^{x/2}$ . Find the smallest n such that the approximation error for the n-th order Taylor polynomial with  $x_0 = 1$  is guaranteed to be less than 0.1 over the interval [-1, 3].

We want  $|f(x) - p_n(x)| = |R_n(x)| \le 0.1$  for all  $x \in [-1,3]$ . Since  $|R_n(x)| \le |B_n(x)|$ , we ensure that  $|B_n(x)| \le 0.1$  for all  $x \in [-1,3]$ . Note that the worst (largest)  $B_n$  always occurs at an endpoint of the interval.

$$B_n(-1) = \max_{\zeta \in [-1,1]} |f^{(n+1)}(\zeta)| \frac{|-1-1|^{n+1}}{(n+1)!}$$

$$= \frac{1}{2^{n+1}} e^{1/2} \frac{2^{n+1}}{(n+1)!}$$

$$= \frac{e^{1/2}}{(n+1)!}.$$

## Example (cont.)

Similarly at x = 3, we have

$$B_n(3) = \max_{\zeta \in [1,3]} |f^{(n+1)}(\zeta)| \frac{|3-1|^{n+1}}{(n+1)!}$$

$$= \frac{1}{2^{n+1}} e^{3/2} \frac{2^{n+1}}{(n+1)!}$$

$$= \frac{e^{3/2}}{(n+1)!}.$$

### Example (cont.)

Again, we tabulate the error bound for each n as follows.

k	$B_n(-1)$	$B_n(3)$
0	1.6487	4.4817
1	0.8244	2.2408
2	0.2748	0.7469
3	0.0687	0.1867
4	0.0137	0.0373

Hence the minimum order n to achieve a precision of 0.1 over [-1,3] is 4.

#### Floating-point operations

The number of floating-point operations (FLOPs) roughly corresponds with the computational cost.

The two common floating-point operations are addition-subtraction "A" and multiplication-division "M".

In many cases, it's possible to optimize away a few FLOPs when evaluating polynomials.

#### Counting FLOPs

For example, to evaluate  $p(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!}$  — a terrible implementation may actually compute

$$x - \frac{x \cdot x \cdot x}{3 \times 2 \times 1} + \frac{x \cdot x \cdot x \cdot x \cdot x}{5 \times 4 \times 3 \times 2 \times 1}$$
 (2A + 14M),

whereas a better implementation may precompute the constants and store  $x^2$  and  $x^3$  as intermediate variables and reuse them.

We'll introduce a general method for evaluating polynomials efficiently next time.