```
> f15b:=sin(ln(x));
```

$$f15b := \sin(\ln(x))$$

```
> f15b1:=diff(f15b,x);
```

$$f15b1 := \frac{\cos(\ln(x))}{x}$$

```
> f15b2:=diff(f15b1,x);
```

$$f15b2 := -\frac{\sin(\ln(x))}{x^2} - \frac{\cos(\ln(x))}{x^2}$$

```
> f15b3:=diff(f15b2,x);
```

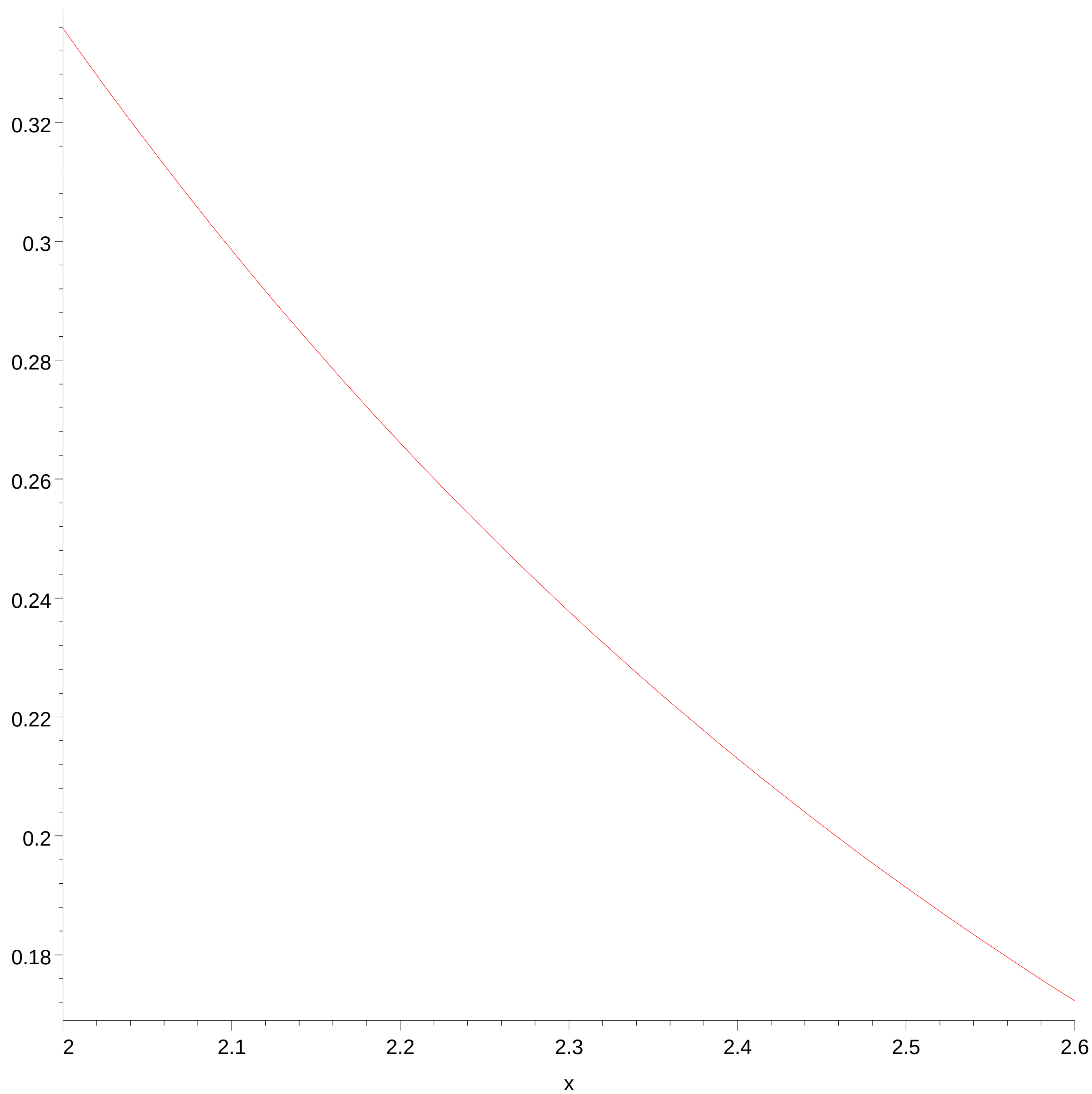
$$f15b3 := \frac{\cos(\ln(x))}{x^3} + 3\,\frac{\sin(\ln(x))}{x^3}$$

```
> f15b4:=diff(f15b3,x);
```

$$f15b4 := -10\,\frac{\sin(\ln(x))}{x^4}$$

The function given in exercise 15b of section 3.1 of Burden & Faires, together with its derivatives. The third derivative is needed for the error estimate, and the fourth derivative gives information that can be used to sharpen the estimate on the third derivative.

```
> plot(f15b3,x=2.0..2.6);
```



This shows function decreasing in agreement with information about its derivative.

```
> dbound:=evalf(subs(x=2.0,f15b3))/6;
```

$$dbound := .05596089022$$

This is the part of the error estimate that comes from the properties of the function.

```
> f15bp:=(x-2.0)*(x-2.4)*(x-2.6);
```

$$f15bp := (x-2.0)(x-2.4)(x-2.6)$$

```
> f15bp1:=diff(f15bp,x);
```

$$f15bp1 := (x-2.4)(x-2.6)+(x-2.0)(x-2.6)+(x-2.0)(x-2.4)$$

```
> maxminloc:=solve(%,x);
```

$$maxminloc := 2.156949913, 2.509716754$$

```
> maxmin:=seq(f15bp,x=[maxminloc]);
```

$$maxmin := .01690089433, -.005049042476$$

Finding the extremes of the polynomial part by calculus.

```
> maxval:=maxmin[1];
```

$$maxval := .01690089433$$

```
> maxval*dbound;
```

$$.0009457890922$$

This is the best that we can prove about the error. The true error is likely to be smaller.

```
> f15bpoly:=subs(x=2.0,f15b)*(x-2.4)*(x-2.6)/(2.0-2.4)/(2.0-2.6)+subs(x=2.4,f15b)*(x-2.0)*(x-2.6)/(2.4-2.0)/(2.4-2.6)+subs(x=2.6,f15b)*(x-2.4)*(x-2.0)/(2.6-2.4)/(2.6-2.0);
```

$$f15bpoly := 4.166666667 \sin(\ln(2.0))\,(x-2.4)\,(x-2.6) - 12.50000000 \sin(\ln(2.4))\,(x-2.0)\,(x-2.6) + 8.333333333 \sin(\ln(2.6))\,(x-2.4)\,(x-2.0)$$

```
> evalf(subs(x=2.1,f15bpoly)),evalf(subs(x=2.1,f15b));%[1]-%[2];
```

$$.6751009587, .6757173135$$

$$-.0006163548$$

```
> evalf(subs(x=2.2,f15bpoly)),evalf(subs(x=2.2,f15b));%[1]-%[2];
```

$$.7086279530, .7092666480$$

$$-.0006386950$$

```
> evalf(subs(x=2.3,f15bpoly)),evalf(subs(x=2.3,f15b));%[1]-%[2];
```

$$.7395422591, .7398915423$$

$$-.0003492832$$

```
> evalf(subs(x=2.5,f15bpoly)),evalf(subs(x=2.5,f15b));%[1]-%[2];
```

$$.7935328070, .7933490026$$

$$.0001838044$$

```
> evalf(subs(x=2.15,f15bpoly)),evalf(subs(x=2.15,f15b));%[1]-%[2];
```

$$.6921910418, .6928744158$$

$$-.0006833740$$

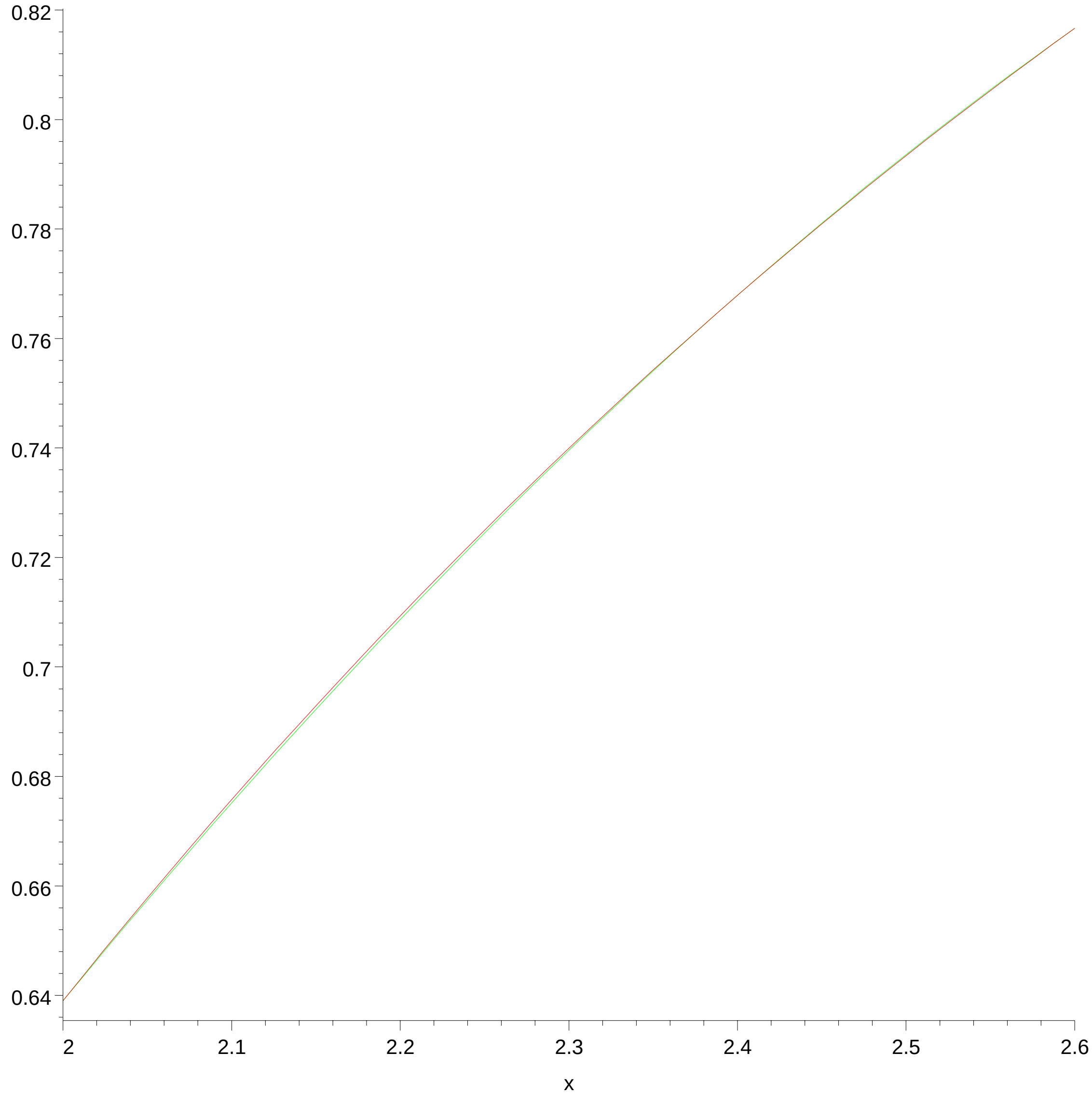
```
> evalf(subs(x=2.16,f15bpoly)),evalf(subs(x=2.16,f15b));%[1]-%[2];
```

$$.6955306778, .6962129281$$

$$-.0006822503$$

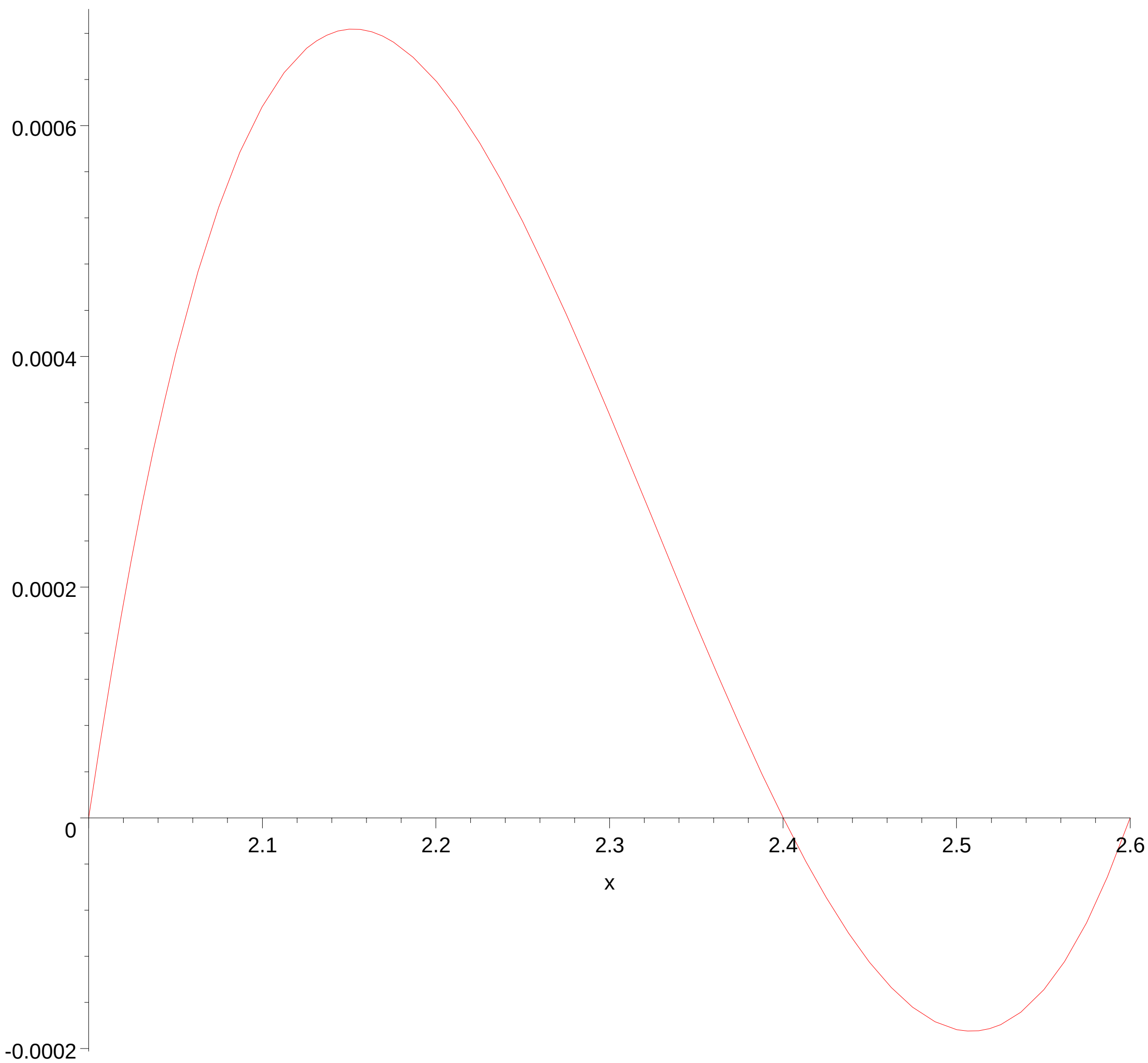
The difference seems largest near the largest value of the function part of the error estimate.

```
> plot({f15b,f15bpoly},x=2.0..2.6);
```



The two curves look close, but visual accuracy here is about 3 decimal places.

```
> plot(f15b-f15bpoly,x=2.0..2.6);
```



Graphing the difference confirms our observations based on selected values.