Taylor's Error

```
> restart:with(plots):with(Student[NumericalAnalysis]):
```

We will approach Taylor's error by approximating the function $f(x) = x\,e^{(x^2)}$ by its fourth degree Taylor polynomial about $x_0 = 0$.

```
> f:=x*exp(x^2);
```

$$f := x\,e^{x^2}$$

By just using x below, it is assumed $x = 0$.

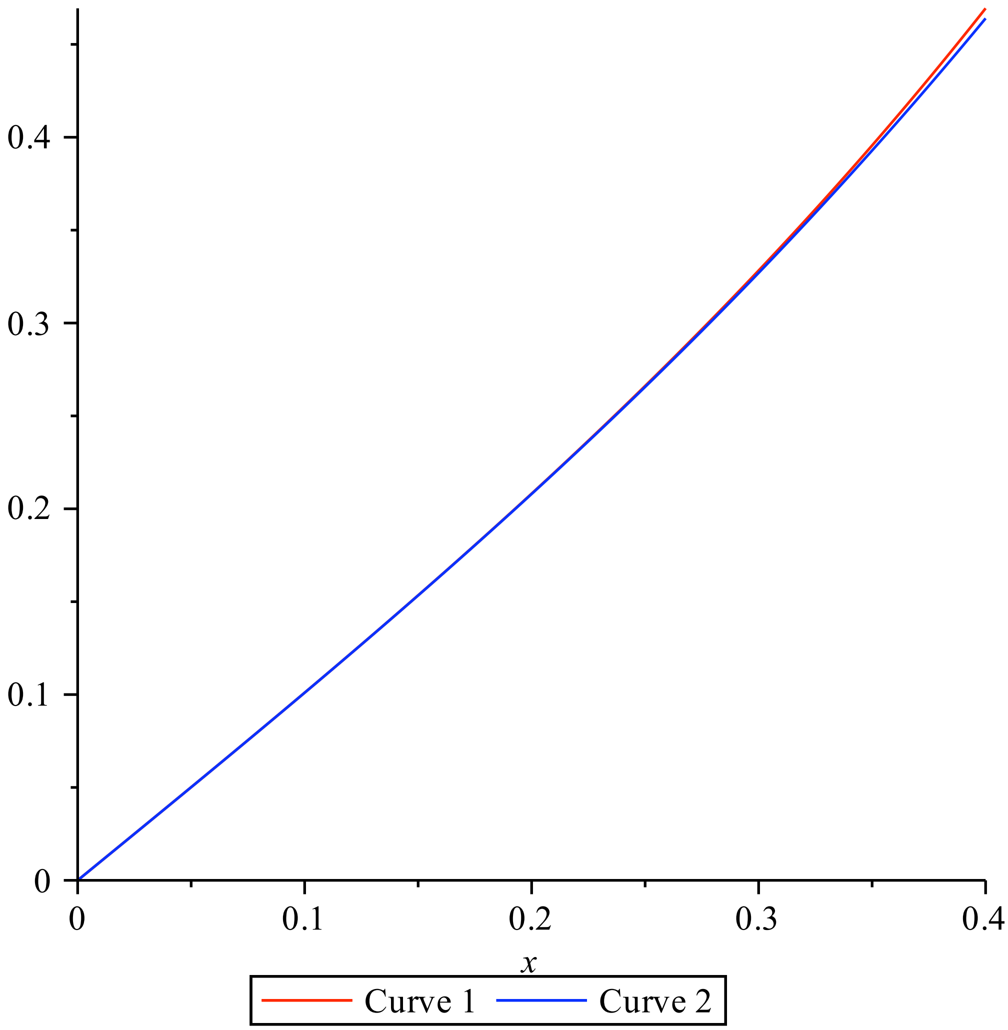
```
> P[4]:=TaylorPolynomial(f,x,order=4,errorboundvar=xi);
```

$$P_4 := \left[x + x^3, \frac{1}{120}\left(60\,e^{\xi^2} + 360\,\xi^2\,e^{\xi^2} + 240\,\xi^4\,e^{\xi^2} + 32\,\xi^6\,e^{\xi^2}\right)x^5\right]$$

The inclusion of the **errorboundvar** causes **TaylorPolynomial** to return a list P[4] where the first element P[4][1] is the Taylor polynomial and the second element P[4][2] is the Taylor error. We can observe that the third and fourth Taylor polynomials are the same here (why?).

We want to find an upper bound for the error $|R_4(x)| = |f(x) - P_{4,1}(x)|$ in using P[4], the 4th degree Taylor polynomial, to approximate f for x = 0 .. 0.4. **<u>This is error over an interval</u>**. We want an **<u>upper bound</u>** for the maximum error at any point from 0 to 0.4. Let's first look at the two graphs.

```
> plot0:=plot(f,x=0..0.4,color=red):
> plot1:=plot(P[4][1],x=0..0.4,color=blue):
> display(plot0,plot1);
```



The graphs look pretty close. Now

$$R_4(x) = \frac{1}{5!} f^{(5)}(\xi(x))\,(x-0)^5$$

where $\xi(x) = 0 .. 0.4$. We have the 5th derivative of f.

```
> f5:=60*exp(xi^2)+360*xi^2*exp(xi^2)+240*xi^4*exp(xi^2)+32*xi^6*
  exp(xi^2);
```

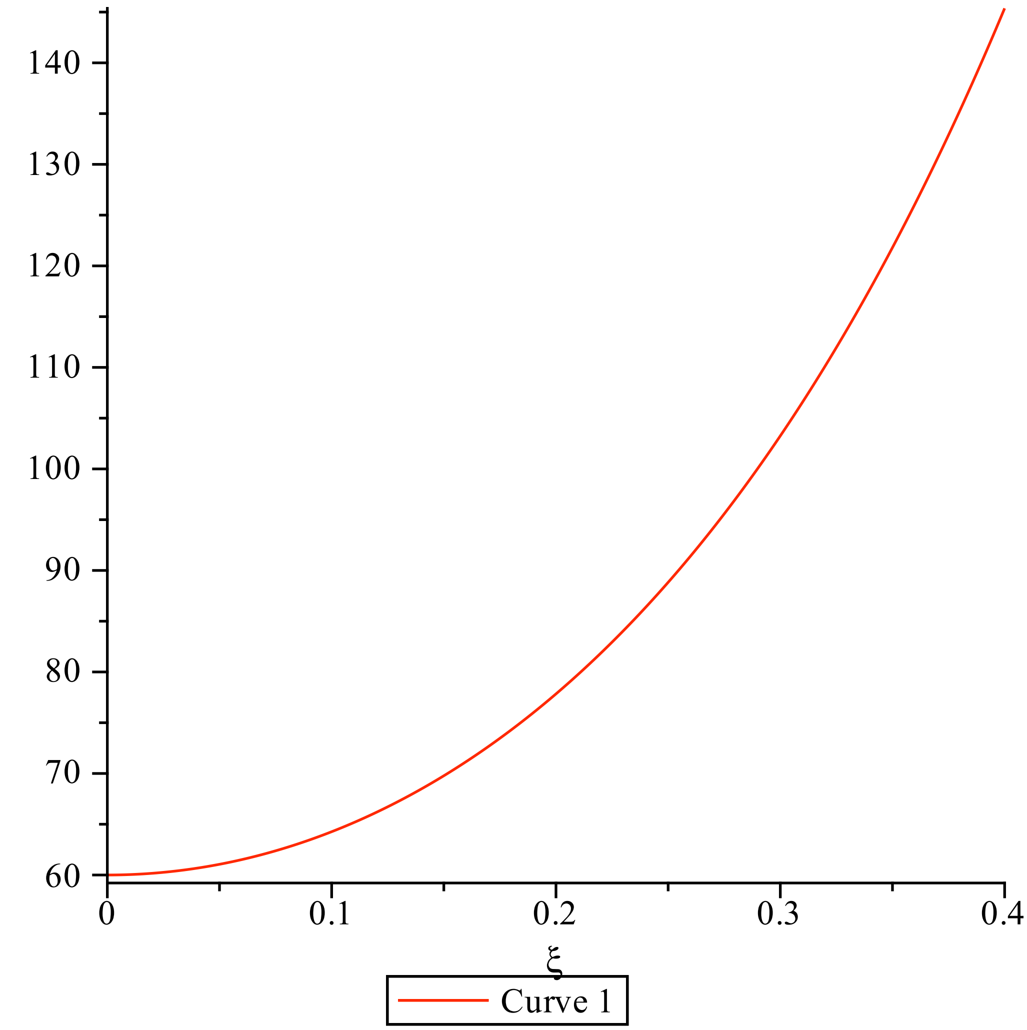
$$f5 := 60\,e^{\xi^2} + 360\,\xi^2\,e^{\xi^2} + 240\,\xi^4\,e^{\xi^2} + 32\,\xi^6\,e^{\xi^2}$$

We now need to find a number M greater than the maximum value of the absolute value of the fifth derivative over the interval. Then

$$|R_4(x)| <= \frac{M}{(4+1)!} \max\left(|x-0|^{(4+1)} \right)$$

for x = 0 .. 0.4. We prefer that this number M be chosen as small as possible. We graph the fifth derivative over the interval.

```
> plot(f5,xi=0..0.4);
```



From the graph it is clear that the absolute value of the fifth derivative never exceeds 150 over our interval, so we choose that number.

```
> maxderiv5:=150;
```

$$maxderiv5 := 150$$

We can also sometimes use the maximize command to find the maximum value of the fifth derivative for us, although this command is very slow to execute.

```
> maximize(abs(f5),x=0..0.4);
```

$$1. \left|60.\, e^{\xi^2} + 360.\, \xi^2 e^{\xi^2} + 240.\, \xi^4 e^{\xi^2} + 32.\, \xi^6 e^{\xi^2}\right|$$

OK, no luck here. But, since the fifth derivative is increasing, this maximum value is just $f5(0.4)$.

```
> evalf(eval(f5,xi=.4));
```

$$145.3687436$$

To be safe, it would be good to round this rumber up a bit (why?), but rarely will this kind of precision be necessary. Choosing 150 will work fine. Now we can find an upper bound for the error over the interval.

```
> maxerror4:=1/120*maxderiv5*0.4^5;
```

$$maxerror4 := 0.01280000000$$

From the graphs above, it appears that the maximum error would occur at $x = 0.4$. Let's find that error.

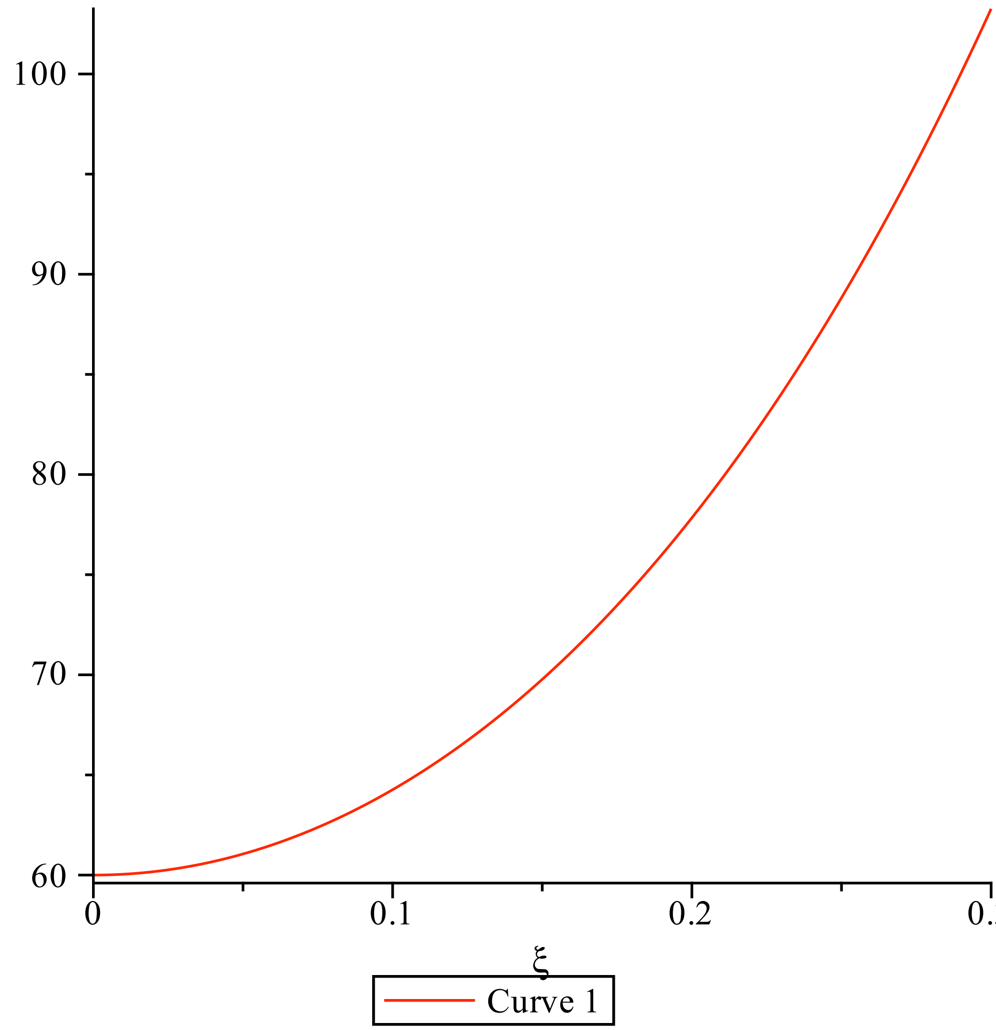
```
> subs(xi=x,f5):
```

```
err:=eval(abs(f-P[4][1]),x=0.4);
```

$$err := 0.0054043484$$

This is certainly less than the upper bound computed. Now suppose we want to find the maximum absolute value for **<u>error at a point</u>**, such as $x = 0.3$. Since ξ is now between 0 and 0.3,we graph the fifth derivative f5 from 0 to 0.3. The other thing that changes from the above is the $x - x_0$, which is simply $0.3 - 0 = 0.3$ here since $x_0 = 0$.

```
> plot(f5,xi=0..0.3);
```



We see from the graph that the absolute value of the fifth derivative never exceeds 105 over this interval, so we choose that number.

```
> maxderiv5:=105;
```

$$maxderiv5 := 105$$

Now we can find an upper bound for the error at the point x=0.3.

```
> maxpointerror4:=1/120*maxderiv5*0.3^5;
```

$$maxpointerror4 := 0.002126250000$$

As expected, this is less than **maxerror4** since **maxerror4** has to take all points between 0 and 0.4 into account, not just 0.3. We can also use the **TaylorPolynomial** command to get this upper bound.

```
> P[4]:=TaylorPolynomial(f,x,order=4,errorboundvar=xi,extrapolate=
  .3);
Warning, unable to compute an upper bound of the error term at x=.3
```

$$P_4 := \left[x + x^3,\ \frac{1}{120}\left(60\,e^{\xi^2} + 360\,\xi^2\,e^{\xi^2} + 240\,\xi^4\,e^{\xi^2} + 32\,\xi^6\,e^{\xi^2}\right) x^5,\ [0.3, 0.327, 0.3282522852]\right]$$

The third item in the list above should contain one more number, namely the upper bound we are searching for. The 0.3 is the value of x, the 0.327 is the Taylor polynomial value at the point, the 0.3282522852 is the function value at the point.

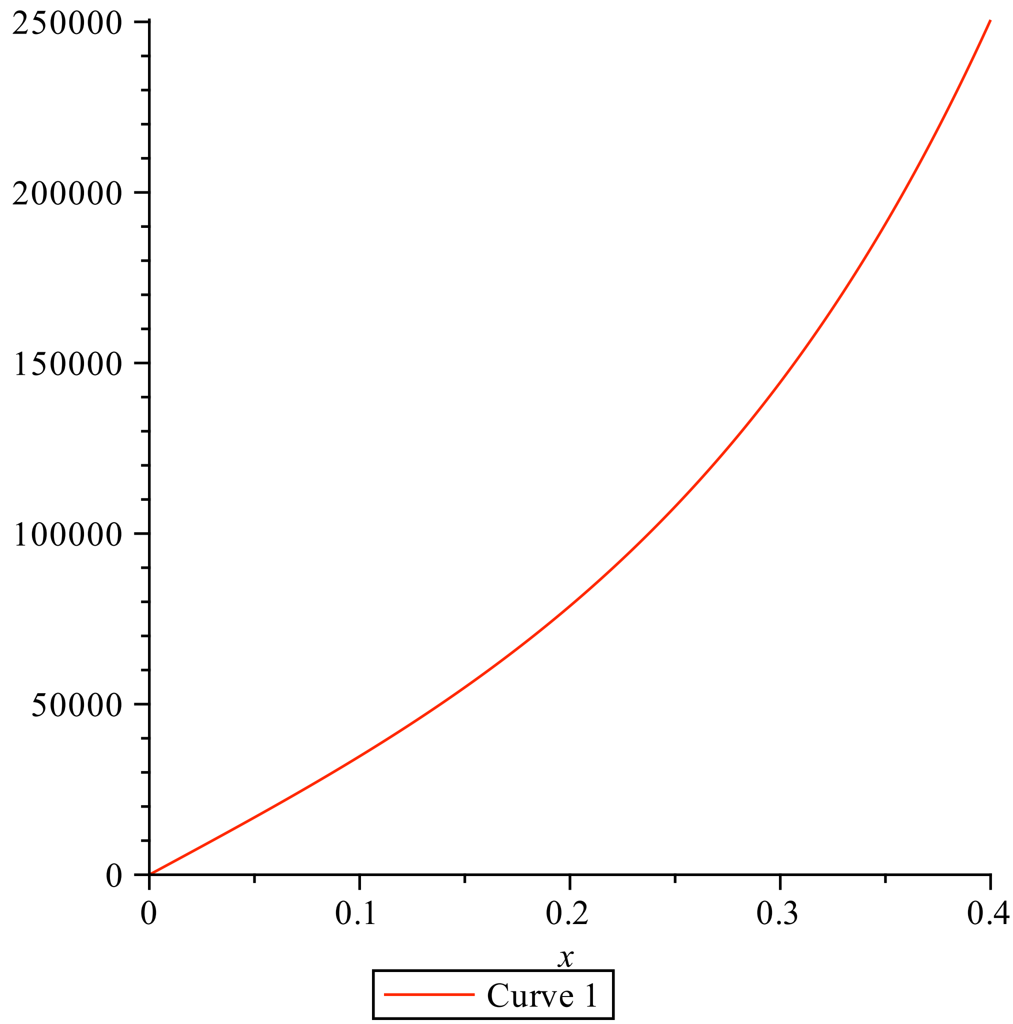
Now suppose that we wish to find a Taylor polynomial that will approximate *f* over this same interval with a maximum error of 0.00001. We need to find the degree *n*. We use a guess and check method. For instance, let's try **order** = 9. We begin by finding the 10th derivative.

```
> f10:=diff(f,x$10);
```

$$f10 := 332640\,x\,e^{x^2} + 1108800\,x^3\,e^{x^2} + 887040\,x^5\,e^{x^2} + 253440\,x^7\,e^{x^2} + 28160\,x^9\,e^{x^2} + 1024\,x^{11}\,e^{x^2}$$

We graph the 10th derivative over our interval.

```
> plot(f10,x=0..0.4);
```



We see that the absolute value of the derivative is always less than 260,000, so we choose this number as maxderiv10.

```
> maxderiv10:=260000;
```

$$maxderiv10 := 260000$$

We now compute the maximum for the absolute value of the error over the interval x = 0 .. 0.4.

```
> maxerror9:=1/10!*maxderiv10*0.4^10;
```

$$maxerror9 := 0.000007512945326$$

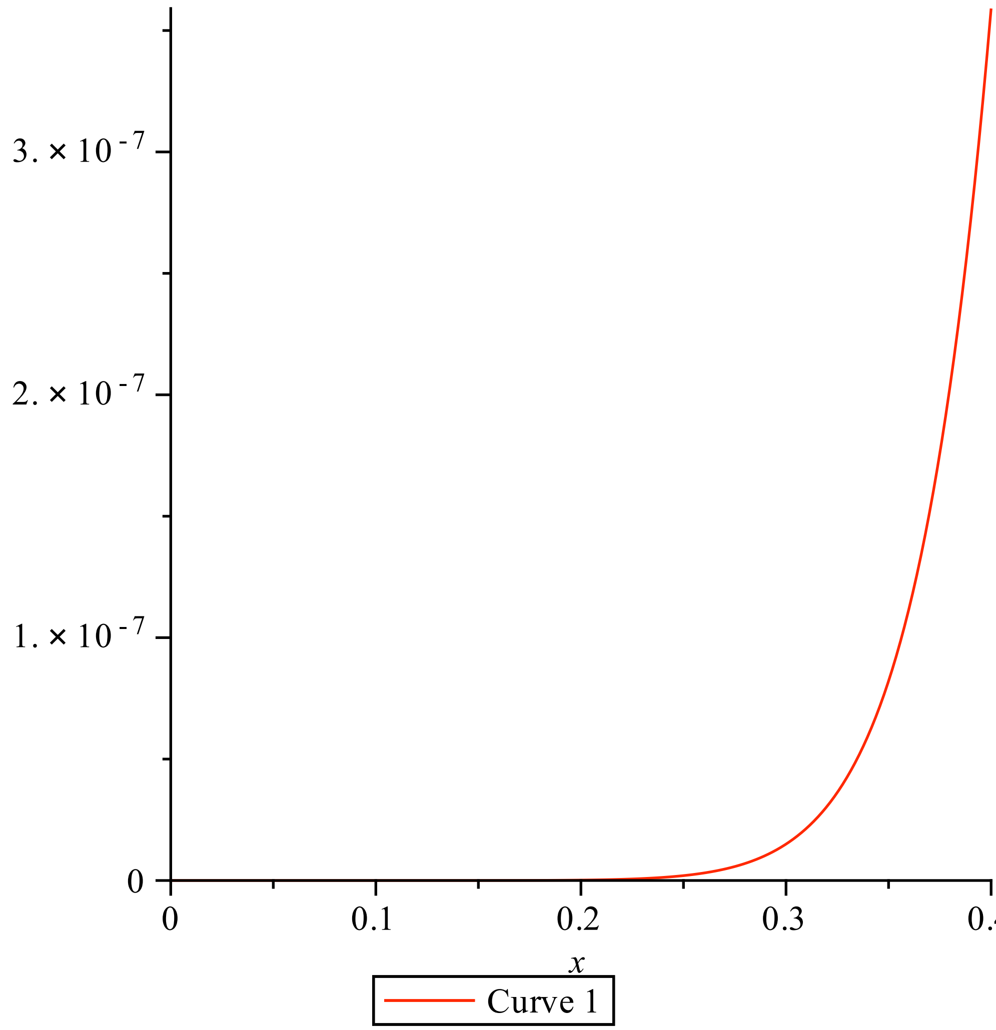
Since this value is less than 0.00001, we choose **order** = 9 and compute the 9th degree Taylor polynomial as our approximation. If our value for maxerror had not been small enough, we would have tried higher values of n until we found one that worked.

```
> P[9]:=TaylorPolynomial(f,x,order=9);
```

$$P_9 := x + x^3 + \frac{1}{2}x^5 + \frac{1}{6}x^7 + \frac{1}{24}x^9$$

We graph the absolute value of the difference $f - P_9$.

```
> plot(abs(f-P[9]),x=0..0.4);
```



It is clear that this difference is everywhere less than 0.00001.

We now wish to approximate $\int_0^{0.4} f(x)\,dx$ by $\int_0^{0.4} P_{4,1}(x)\,dx$. We use the int command for integration.

First, let's find the actual value of the integral.

```
> actualint:=int(f,x=0..0.4);
```

$$actualint := 0.08675543550$$

Now the approximated integral.

```
> approxint:=int(P[4][1],x=0..0.4);
```

$$approxint := 0.08640000000$$

We compute the absolute error.

```
> interror:=abs(actualint-approxint);
```

$$interror := 0.00035543550$$

We can compute an upper bound for this error by integrating maxerror4 that we found above.

```
> maxinterror4:=int(maxerror4,x=0..0.4);
```

$$maxinterror4 := 0.005120000000$$

Finally, we approximate $f\,'(.2)$ by $P_{4,\,1}\,'(.2)$. First, the actual value.

```
> actualderiv:=eval(diff(f,x),x=.2);
```

$$actualderiv := 1.124075636$$

Then the approximation.

```
> approxderiv:=eval(diff(P[4][1],x),x=.2);
```

$$approxderiv := 1.12$$

We compute this absolute error also.

```
> deriverror:=abs(actualderiv-approxderiv);
```

$$deriverror := 0.004075636$$

Not bad!