Composite Numerical Integration

```
> restart;
> with(plots):
```

Many of the standard numerical integration techniques are implemented in Maple in the Student[NumericalAnalysis] package.

```
> with(Student[NumericalAnalysis]);
```

[*AbsoluteError*, *AdamsBashforth*, *AdamsBashforthMoulton*, *AdamsMoulton*, *AdaptiveQuadrature*, *AddPoint*, *ApproximateExactUpperBound*, *ApproximateValue*, *BackSubstitution*, *BasisFunctions*, *Bisection*, *CubicSpline*, *DataPoints*, *Distance*, *DividedDifferenceTable*, *Draw*, *Euler*, *EulerTutor*, *ExactValue*, *FalsePosition*, *FixedPointIteration*, *ForwardSubstitution*, *Function*, *InitialValueProblem*, *InitialValueProblemTutor*, *Interpolant*, *InterpolantRemainderTerm*, *IsConvergent*, *IsMatrixShape*, *IterativeApproximate*, *IterativeFormula*, *IterativeFormulaTutor*, *LeadingPrincipalSubmatrix*, *LinearSolve*, *LinearSystem*, *MatrixConvergence*, *MatrixDecomposition*, *MatrixDecompositionTutor*, *ModifiedNewton*, *NevilleTable*, *Newton*, *NumberOfSignificantDigits*, *PolynomialInterpolation*, *Quadrature*, *RateOfConvergence*, *RelativeError*, *RemainderTerm*, *Roots*, *RungeKutta*, *Secant*, *SpectralRadius*, *Steffensen*, *Taylor*, *TaylorPolynomial*, *UpperBoundOfRemainderTerm*, *VectorLimit*]

We consider a simple example to illustrate the various methods.

```
> f:=x^4;
```

$$f := x^4$$

```
> Int(f,x=0..6)=int(f,x=0..6);
```

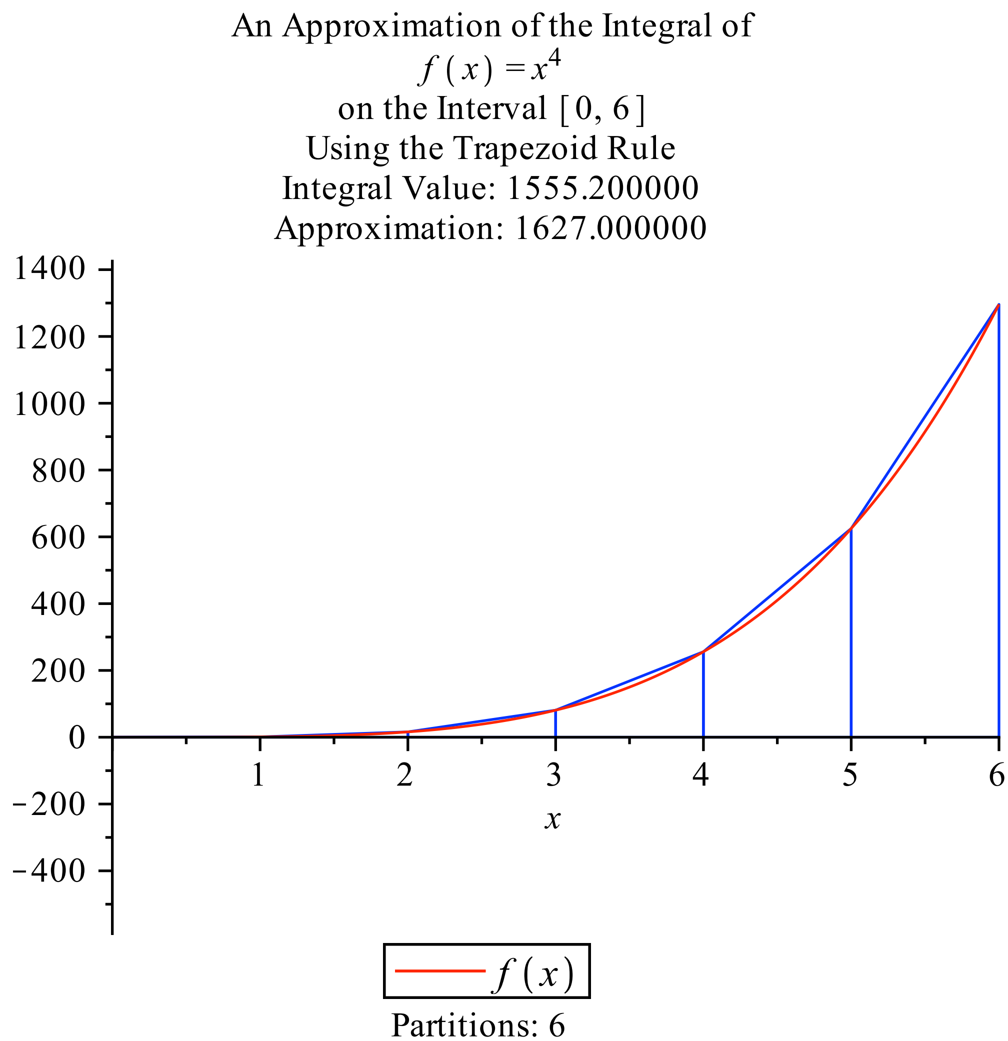
$$\int_0^6 x^4 \, dx = \frac{7776}{5}$$

```
> evalf(rhs(%));
```

$$1555.200000$$

We use the Quadrature command from the NumericalAnalysis package for numerical integration. The third argument gives the **method**. The fourth argument, the **number of partitions**, is the number of times the basic Newton-Coates interval (h for trapezoid and 2h for simpson and midpoint) fits in the interval of integration. This is n for the composite trapezoid, $\frac{n}{2}$ for the composite Simpson's methods, and $\frac{n}{2} + 1$ for the composite midpoint method. Recall that n must be even for the composite Simpson's and composite midpoint methods. The output argument is optional, with the default being "output=value."

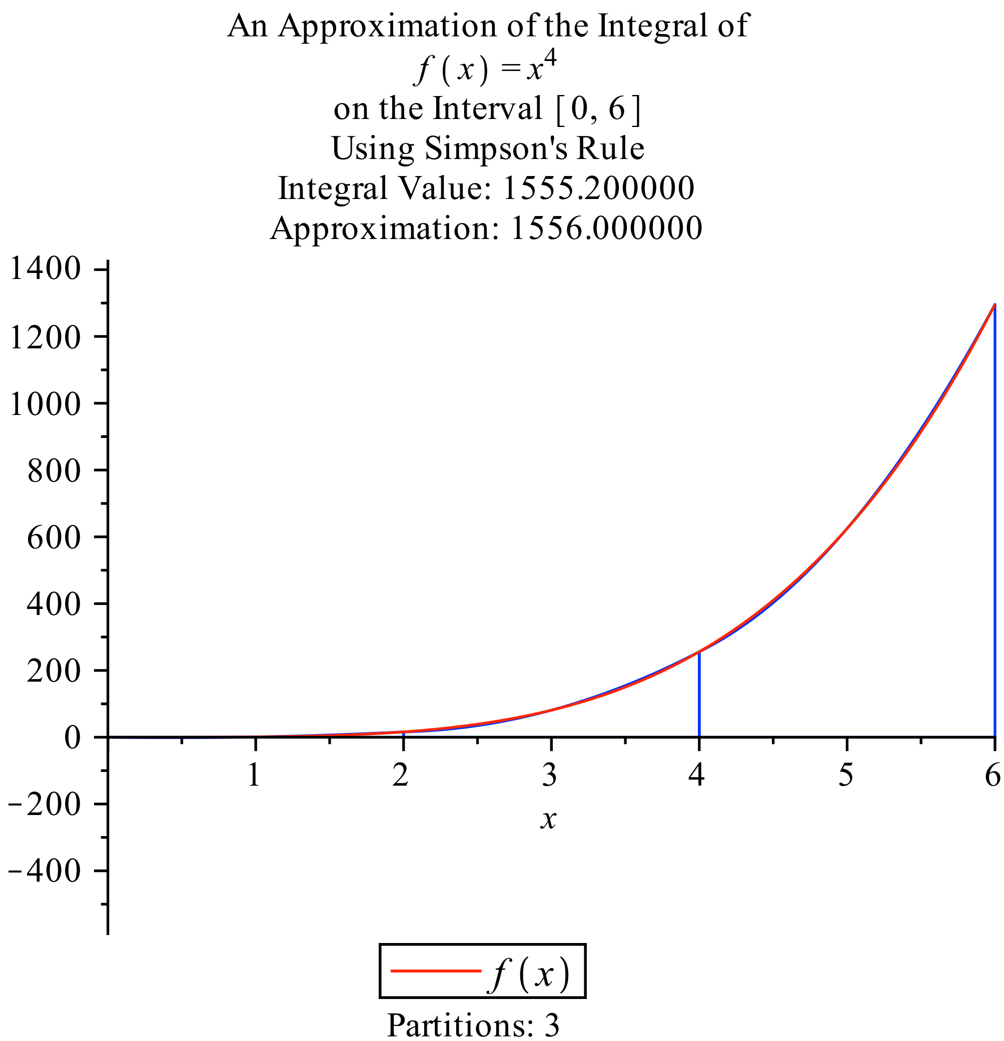
```
> Quadrature(f,x=0..6,method=trapezoid,partition=6,output=plot);
```




```
> Quadrature(f,x=0..6,method=trapezoid,partition=6,output=sum);
```

$$\frac{1}{2}\sum_{i=0}^{5}\left(i^4 + (i+1)^4\right)$$

```
> Quadrature(f,x=0..6,method=trapezoid,partition=6,output=
  information);
INTEGRAL: Int(x^4,x=0..6) = 1555.2
APPROXIMATION METHOD: Trapezoidal Rule
---------------------------------- INFORMATION TABLE -------------------
----------------
    Approximate Value          Absolute Error          Relative Error
                1627                     71.8                4.617 %
------------------------------------------------------------------------
----------------
Number of Function Evaluations:      7
> Quadrature(f,x=0..6,method=simpson,partition=6/2,output=plot);
```

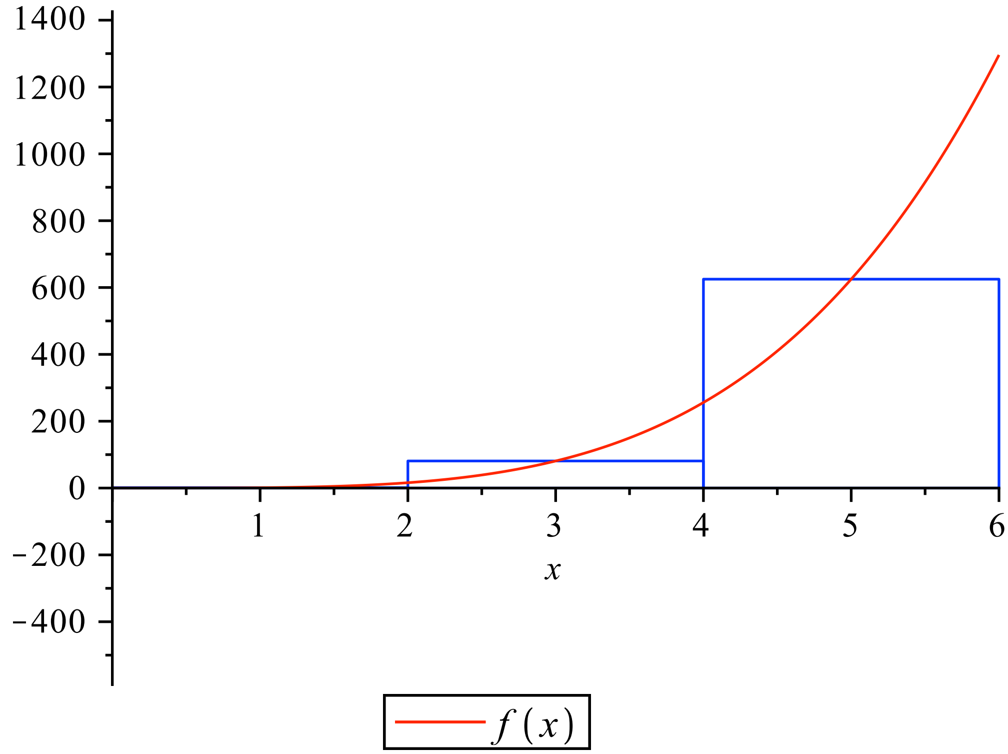



```
> Quadrature(f,x=0..6,method=simpson,partition=6/2,output=sum);
```

$$\frac{1}{3} \sum_{i=0}^{2} \left(16\, i^4 + 4\,(2\,i+1)^4 + (2\,i+2)^4\right)$$

```
> Quadrature(f,x=0..6,method=simpson,partition=6/2,output=
  information);
INTEGRAL: Int(x^4,x=0..6) = 1555.2
APPROXIMATION METHOD: Simpson's Rule
---------------------------------- INFORMATION TABLE -------------------
----------------
    Approximate Value          Absolute Error          Relative Error
                 1556                     0.8              0.05144 %
------------------------------------------------------------------------
----------------
Number of Function Evaluations:      7
> Quadrature(f,x=0..6,method=midpoint,partition=4/2+1,output=plot);
```

An Approximation of the Integral of
$f(x) = x^4$
on the Interval $[0, 6]$
Using a Midpoint Riemann Sum
Integral Value: 1555.200000
Approximation: 1414.000000



Partitions: 3

```
> Quadrature(f,x=0..6,method=midpoint,partition=4/2+1,output=sum);
```

$$2\left(\sum_{i=0}^{2}(2i+1)^4\right)$$

```
> Quadrature(f,x=0..6,method=midpoint,partition=4/2+1,output=
  information);
INTEGRAL: Int(x^4,x=0..6) = 1555.2
APPROXIMATION METHOD: Midpoint Rule
---------------------------------- INFORMATION TABLE ------------------
---------------
    Approximate Value          Absolute Error          Relative Error
            1414                         141.2                 9.079 %
-----------------------------------------------------------------------
---------------
Number of Function Evaluations:      3
```

Page 128, Problem 6

In this problem we determine the values of n and h necessary to approximate $\int_0^{\pi} x^2 \cos(x)\, dx$ to within $10^{(-4)}$ by the Composite Trapezoidal, Composite Simpson's, and Midpoint Rules.

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

$$f := x^2 \cos(x)$$

We first find the exact value of the integral for comparison.

```
> exact:=int(f,x=0..Pi);
```

$$exact := -2\pi$$

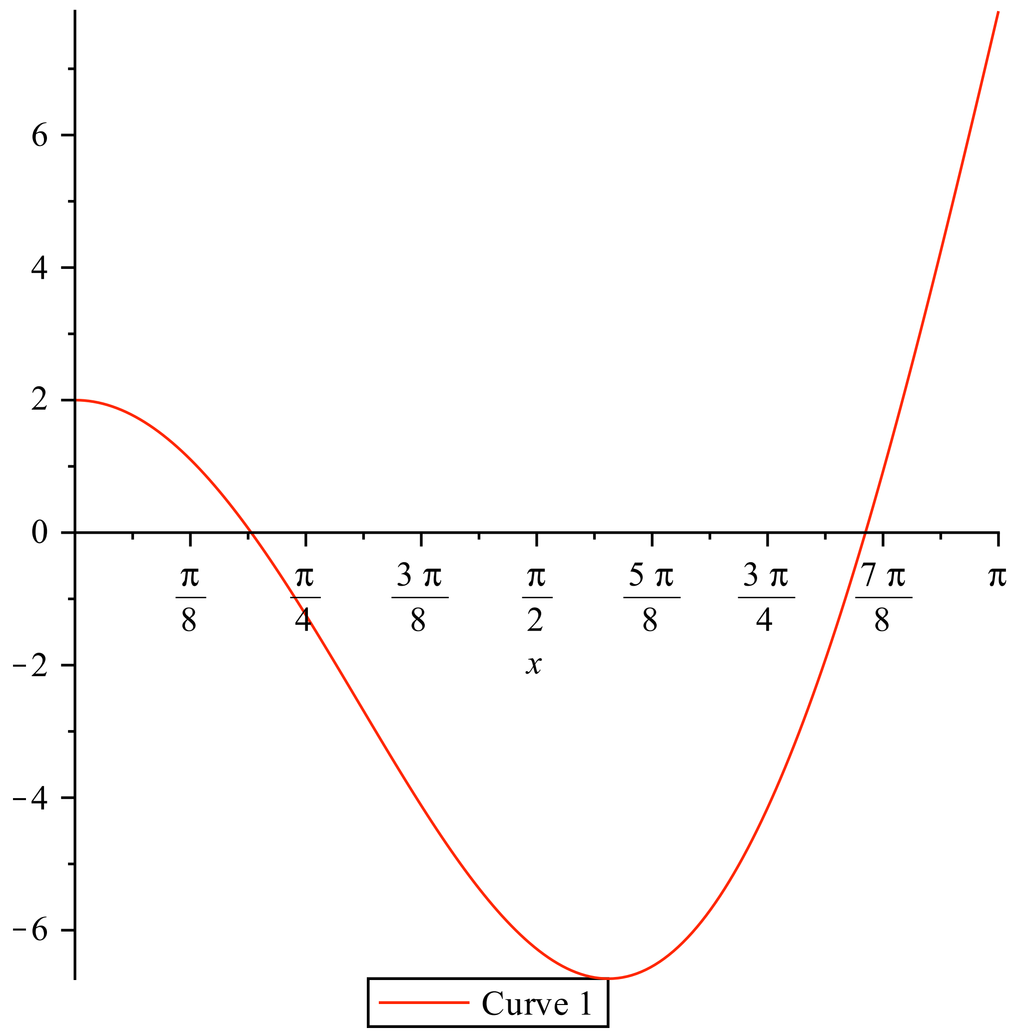
Composite Trapezoidal

The error term for the Composite Trapezoidal Rule is $-\frac{b-a}{12}h^2 f''(\mu)$ where μ is between a and b. We determine first n, then h.

```
> f2:=diff(f,x$2);
```

$$f2 := 2\cos(x) - 4x\sin(x) - x^2\cos(x)$$

```
> plot(f2,x=0..Pi);
```



The maximum of the absolute value of this second derivative is less than 8.

```
> maxderiv2:=8;
```

$$maxderiv2 := 8$$

We find an upper bound for the error.

```
> maxerrortrap:=Pi/12*(Pi/n)^2*maxderiv2;
```

$$maxerrortrap := \frac{2}{3} \frac{\pi^3}{n^2}$$

We set this maximum error equal to $10^{(-4)}$ and solve for n.

```
> fsolve(maxerrortrap=10^(-4),n=0..infinity);
```

$$454.6520771$$

We set $n = 455$ and use that value to get our h.

```
> n:=455;h:=Pi/n;
```

$$n := 455$$

$$h := \frac{1}{455} \pi$$

We now approximate the integral using the **Composite Trapezoidal Rule** and find the error in the approximation.

```
> trap:=evalf(Quadrature(f,x=0..Pi,method=trapezoid,partition=n));
```

$$trap := -6.283210265$$

We find the actual absolute error.

```
> trap_error:=evalf(abs(exact-trap));
```

$$trap\_error := 0.000024957$$

We compare this to our upper absolute error bound.

```
> maxerrortrap:=evalf(maxerrortrap);
```

$$maxerrortrap := 0.00009984712534$$

Composite Simpson.

The error term for the Composite Simpson's Rule is $-\frac{b-a}{180} h^4 f^{(4)}(\mu)$, where μ is between a and b. We determine first n, which must be even, then h. We first reset n.

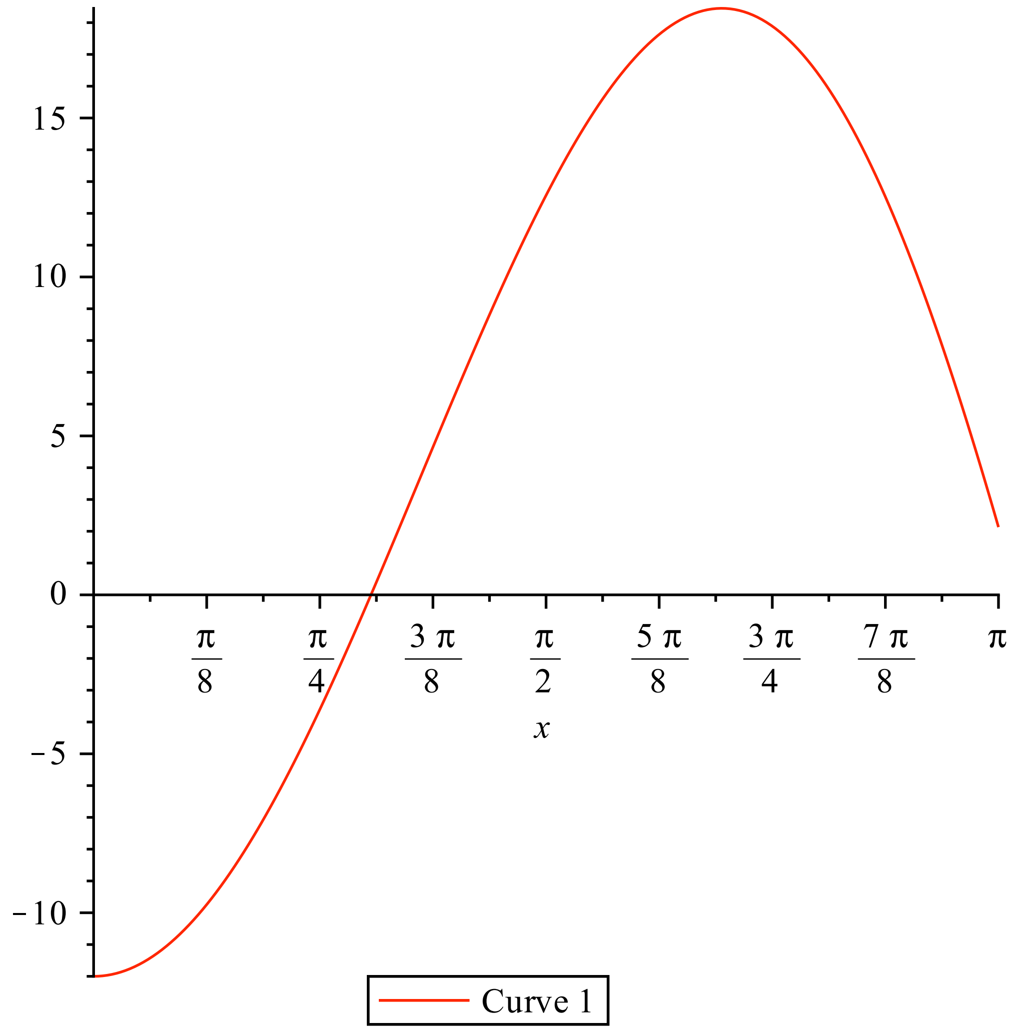
```
> n:='n';
```

$$n := n$$

```
> f4:=diff(f,x$4);
```

$$f4 := -12 \cos(x) + 8 x \sin(x) + x^2 \cos(x)$$

```
> plot(f4,x=0..Pi);
```



The maximum of the absolute value of this fourth derivative is less than 19.

```
> maxderiv4:=19;
```

$$maxderiv4 := 19$$

We find an upper bound for the error.

```
> maxerrorsimp:=Pi/180*(Pi/n)^4*maxderiv4;
```

$$maxerrorsimp := \frac{19}{180} \frac{\pi^5}{n^4}$$

We set this maximum error equal to $10^{(-4)}$ and solve for n.

```
> fsolve(maxerrorsimp=10^(-4),n=0..infinity);
```

$$23.84007486$$

We set $n = 24$ and use that value to get our h.

```
> n:=24;h:=Pi/n;
```

$$n := 24$$

$$h := \frac{1}{24} \pi$$

We now approximate the integral using the **Composite Simpson's Rule** and find the error in the approximation.

```
> simp:=evalf(Quadrature(f,x=0..Pi,method=simpson,partition=n/2));
```
$$simp := -6.283154461$$

We find the actual absolute error.

```
> simp_error:=evalf(abs(exact-simp));
```
$$simp\_error := 0.000030847$$

We compare this to our upper absolute error bound.

```
> maxerrorsimp:=evalf(maxerrorsimp);
```
$$maxerrorsimp := 0.00009736110465$$

Composite Midpoint

The error term for the Composite Midpoint rule is $\frac{b-a}{6}\,h^2 f''(\mu)$ where μ is between a and b. We first determine *n*, where *n* must be even, then *h*. We again reset *n*.

```
> n:='n';
```
$$n := n$$

We find an upper bound for the error.

```
> maxerrormid:=Pi/6*(Pi/(n+2))^2*maxderiv2;
```
$$maxerrormid := \frac{4}{3}\,\frac{\pi^3}{(n+2)^2}$$

We set this maximum error equal to $10^{(-4)}$ and solve for *n*.

```
> fsolve(maxerrormid=10^(-4),n=0..infinity);
```
$$640.9751336$$

We set $n = 642$ and use that value to get our *h*.

```
> n:=642;h:=Pi/(n+2);
```
$$n := 642$$
$$h := \frac{1}{644}\,\pi$$

We now approximate the integral using the **Composite Midpoint Rule** and find the error in the approximation.

```
> mid:=evalf(Quadrature(f,x=0..Pi,method=midpoint,partition=n/2+1));
```
$$mid := -6.283160396$$

We find the actual absolute error.

```
> mid_error:=evalf(abs(exact-mid));
```
$$mid\_error := 0.000024912$$

We compare this to our upper absolute error bound.

```
> maxerrormid:=evalf(Pi/6*(Pi/(n+2))^2*maxderiv2);
```
$$maxerrormid := 0.00009968197180$$