A2Q10

Repeat what you did in Question 9 for Simpson's rule. Recall that Simpson's rule Sn where n is even is given by

$$Sn = \frac{h}{3} \cdot (f(a) + 4 \cdot f(a+h) + 2 \cdot f(a+2 \cdot h) + 4 \cdot f(a+3 \cdot h) + 2 \cdot f(a+4 \cdot h) + \ldots + 4 \cdot f(a+(n-1) \cdot h) + f(a+n \cdot h))$$

Notice the pattern of the coefficients. For n=8 they are 1 4 2 4 2 4 2 4 1.

Your Maple procedure SimpsonsRule(f(x),x,a,b,n) should get the values below.

You should find that Simpson's rule is more accurate than the Midpoint rule.

In one sentence, state how much more accurate Simpson's rule is than the Midpoint rule.

```
> restart:
> SimpsonsRule := proc(f,x::name,a::numeric,b::numeric,n::numeric)

      Digits := 15;
      local h,g,w,i,S;

      if n = 0 then
          return 0;
      fi;

      h := (b-a)/n;
      for i from 0 to n do
          if i = 0 then
              w := a;
              g := evalf[Digits](eval(f,x=w));
          elif i = n then
              w := w + h;
              g := g + evalf[Digits](eval(f,x=w));
          elif (i mod 2 = 1) then
              w := w + h;
              g := g + 4*evalf[Digits](eval(f,x=w));
          else
              w := w + h;
              g := g + 2*evalf[Digits](eval(f,x=w));
          fi;
      od:
      S := evalf((h/3)*g):

      end:
> SimpsonsRule(sin(x),x,0,1,4);
  SimpsonsRule(sin(x),x,0,1,8);
  SimpsonsRule(sin(x),x,0,1,16);
```

0.459707744927313

0.459698318798463

$$0.459697733119047 \tag{1}$$

Compute the error in these approximations

```
> Digits := 15;
```

$$Digits := 15 \tag{2}$$

```
> A := evalf( int(sin(x), x=0..1) );
```

$$A := 0.459697694131860 \tag{3}$$

```
> for n in [4,8,16,32,64,128] do
      Sn := SimpsonsRule(sin(x),x,0,1,n);
      e[n] := abs(A-Sn);
      printf("n=%3d error=%.15f\n", n, e[n]);
  od:
n=  4 error=0.000010050795453
n=  8 error=0.000000624666603
n= 16 error=0.000000038987187
n= 32 error=0.000000002435850
n= 64 error=0.000000000152227
n=128 error=0.000000000009510
```

```
> 0.000000624666603 / 0.000010050795453;
```

$$0.0621509616747346 \tag{4}$$

```
> e[16]/e[32];
```

$$16.0055779296755 \tag{5}$$

```
> e[32]/e[64];
```

$$16.0014320718401 \tag{6}$$

If we compare the e[n]/e[2n] we see that the error is dropping by a factor of 16.