Subject : The following demonstrates the convergence of the Trapezoidal method of estimating integrals of continuous functions.
Authors : Nathan Collier, Autar Kaw, Loubna Guennoun
Date : 27 October 2005

■ Introduction

Simpson's rule is based on the Newton-Cotes formula that if one approximates the integrand of the integral by an n^{th} order polynomial, then the integral of the function is approximated by the integral of that n^{th} order polynomial. Integration of polynomials is simple and is based on the calculus. Simpson's $1/3^{rd}$ rule is the area under the curve where the function is approximated by a second order polynomial. [click **here** for textbook notes] [click **here** for power point presentation].

The following simulation illustrates the convergence of Simpson's $1/3^{rd}$ rule of integration. This section is the only section where the user interacts with the program. The user enters any function *f(x)*, the lower and upper limit of the integration, and the maximum number of segments, *n*. The program will display the true error, the absolute relative percentage true error, the approximate error, the absolute relative percentage approximate error, and the least number of significant digits correct in the approximation.

■ Inputs

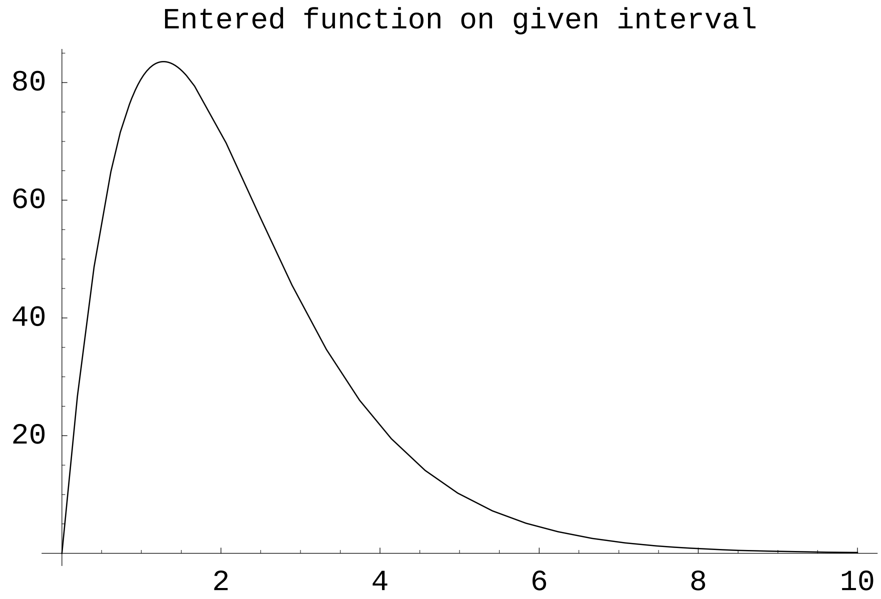
Integrand $f[x] = 0$

```
In[1]:= f[x_] := 300.0 * x / (1.0 + Exp[x]);
```

Lower and upper limit of the integral, *a* and *b* respectively.

```
In[2]:= a = 0.0;
        b = 10.0;
```

In[4]:=
```
curve = Plot[f[x], {x, a, b}, PlotLabel → "Entered function on given interval",
    TextStyle → {FontSize → 11}];
General::spelll Off;
```



Maximum number of segments. Note that this number must be even.

In[6]:=
```
n_maximum = 40;
```

■ True Solution

This is the solution found by Mathematica

In[7]:=
```
Actual = Integrate[f[x], {x, a, b}]
```

Out[7]= 246.59

■ Value of integral as a function of segments

The following procedure determines the value of the integral from *a* to *b* using Simpson's 1/3rd rule with *n* segments.

In[8]:=
```
Array[AV, n_maximum];
```

In[9]:=
```
For[i = 2, i ≤ n_maximum, i = i + 2,
  h = (b - a) / i;
  sum1 = Sum[f[a + jj * h], {jj, 1, i - 1, 2}];
  sum2 = Sum[f[a + jj * h], {jj, 2, i - 2, 2}];
  AV[i] = h / 3 * (f[a] + 4 * sum1 + 2 * sum2 + f[b])]
```

■ Absolute true error

In[10]:=
```
Array[Et, n_maximum];
```

In[11]:=
```
For[i = 2, i <= n_maximum, i = i + 2, Et[i] = Abs[Actual − AV[i]]]
```

■ Absolute relative true error

In[12]:=
```
Array[∈t, n_maximum];
```

In[13]:=
```
For[i = 2, i <= n_maximum, i = i + 2, ∈t[i] = Abs[Et[i] / Actual * 100]]
```

■ Absolute approximate error

In[14]:=
```
Array[Ea, n_maximum];
```

In[33]:=
```
For[i = 2, i <= n_maximum, i = i + 2,
 If[i ≤ 2, Ea[i] = 0, Ea[i] = Abs[AV[i] − AV[i − 2]]]]
```

■ Absolute relative approximate error

In[34]:=
```
Array[∈a, n_maximum];
```

In[35]:=
```
For[i = 2, i <= n_maximum, i = i + 2,
 If[i ≤ 2, ∈a[i] = 0, ∈a[i] = Abs[Ea[i] / AV[i] * 100]]]
```

■ Significant digits at least correct

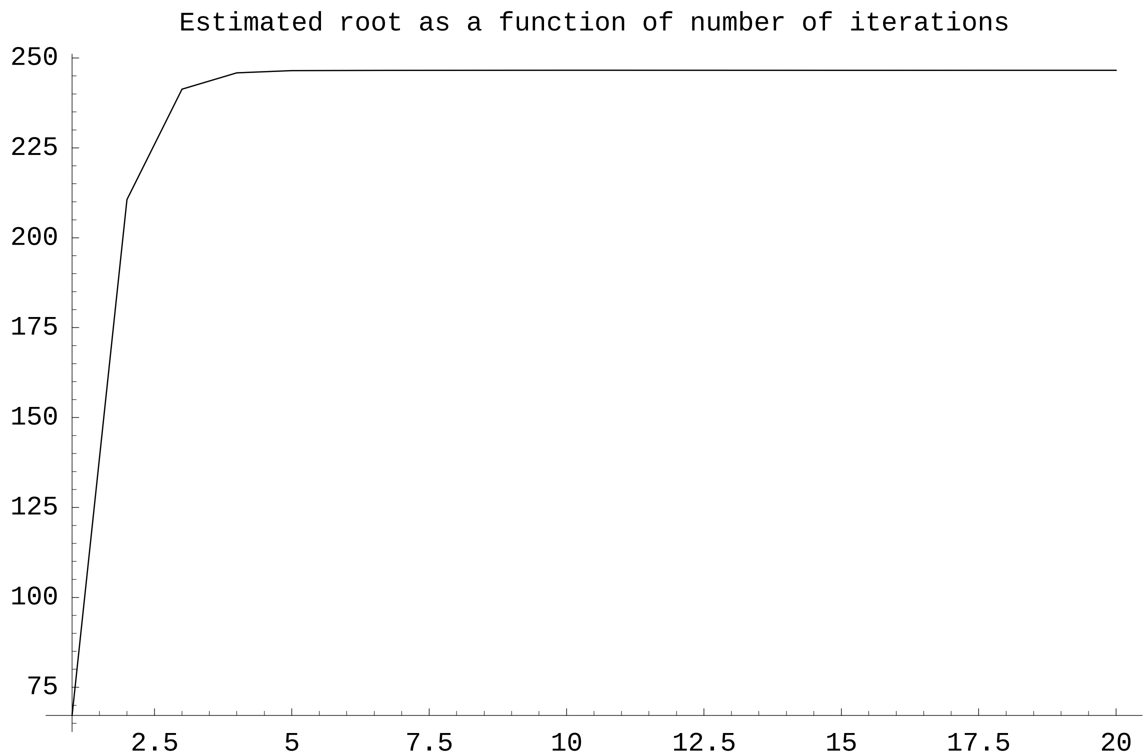
In[36]:=
```
Array[sigdig, n_maximum];
```

In[37]:=
```
For[i = 2, i <= n_maximum, i = i + 2, If[(∈a[i] ≥ 5) || (i ≤ 2),
  sigdig[i] = 0, sigdig[i] = Floor[(2 − Log[10, Abs[∈a[i] / 0.5]])]]]
```

■ Graphs

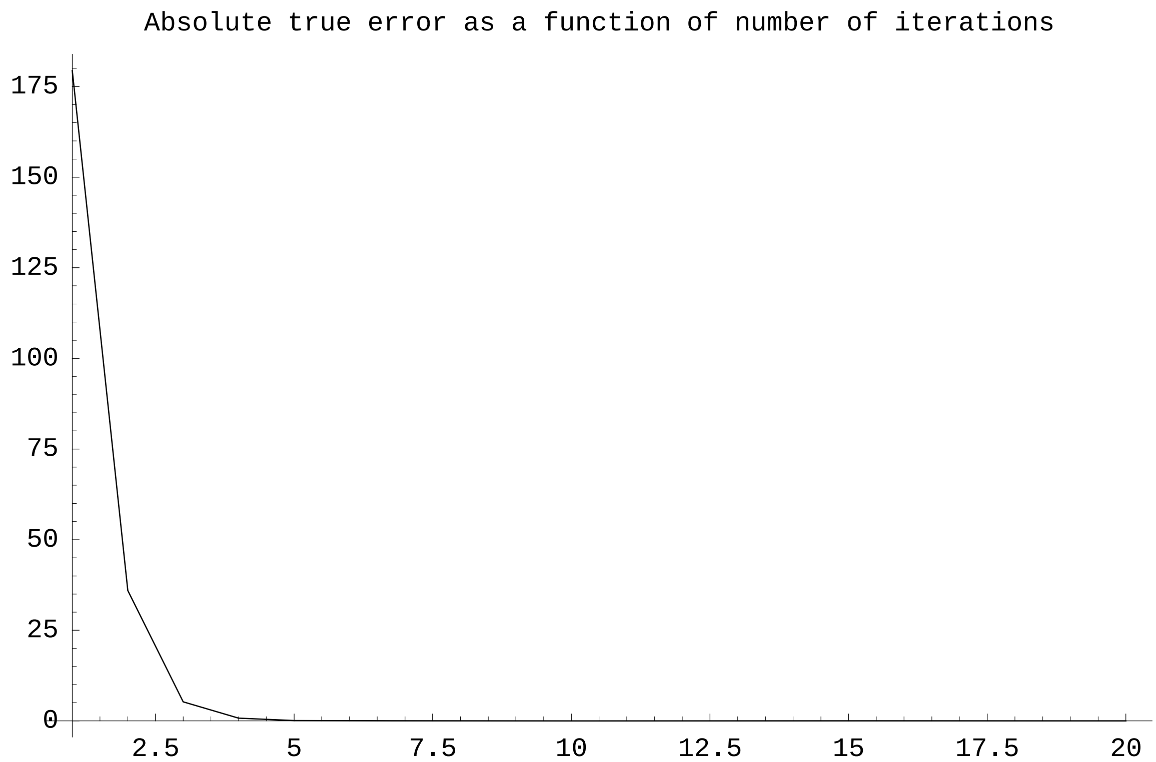
In[20]:=
```
xrplot = Table[AV[i], {i, 2, n_maximum, 2}];
```

```
In[21]:= ListPlot[xrplot, PlotJoined → True,
          PlotRange -> All, AxesOrigin -> {1, Min[xrplot]},
          PlotLabel -> "Estimated root as a function of number of iterations"];
```



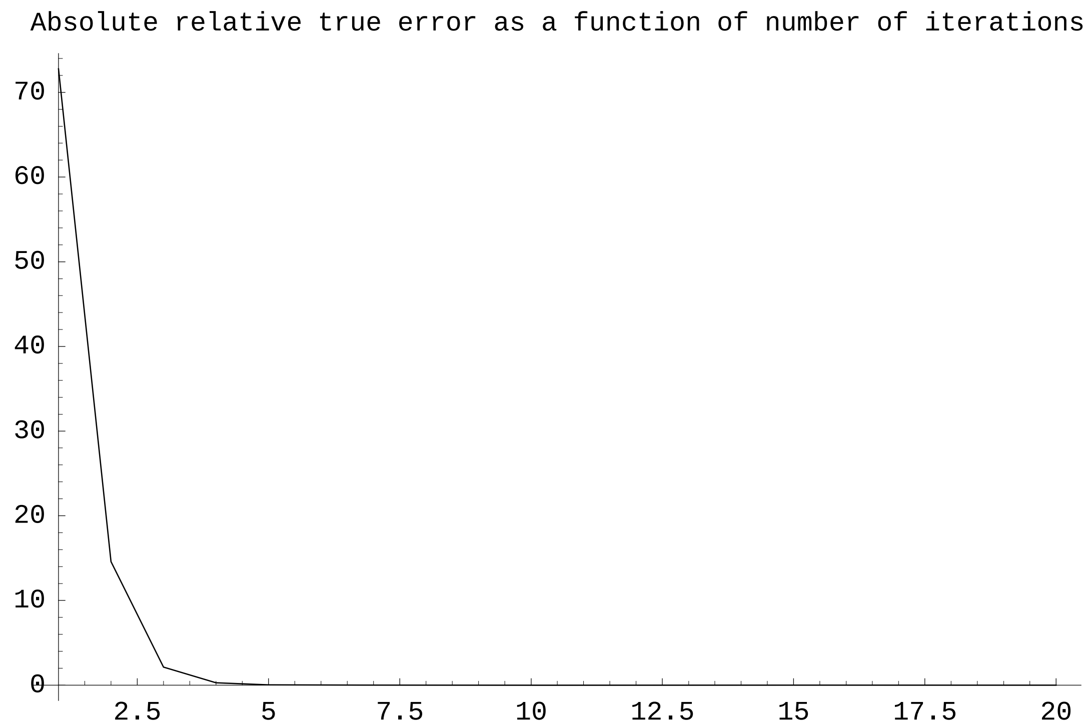

```
In[22]:= Etplot = Table[Et[i], {i, 2, n_maximum, 2}];
```

In[23]:=
```
ListPlot[Etplot, PlotJoined → True,
 PlotRange -> All, AxesOrigin -> {1, Min[Etplot]},
 PlotLabel -> "Absolute true error as a function of number of iterations"];
```



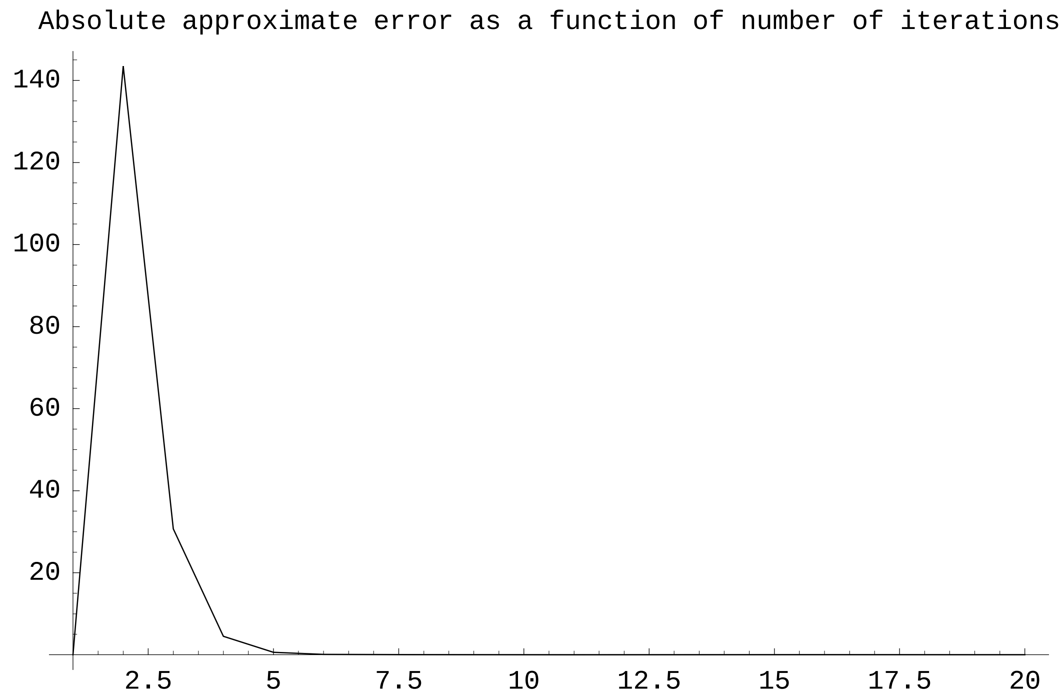
In[24]:=
```
ϵtplot = Table[ϵt[i], {i, 2, n_maximum, 2}];
```

In[25]:=
```
ListPlot[∈tplot, PlotJoined → True,
 PlotRange -> All, AxesOrigin -> {1, Min[∈tplot]}, PlotLabel ->
  "Absolute relative true error as a function of number of iterations"];
```



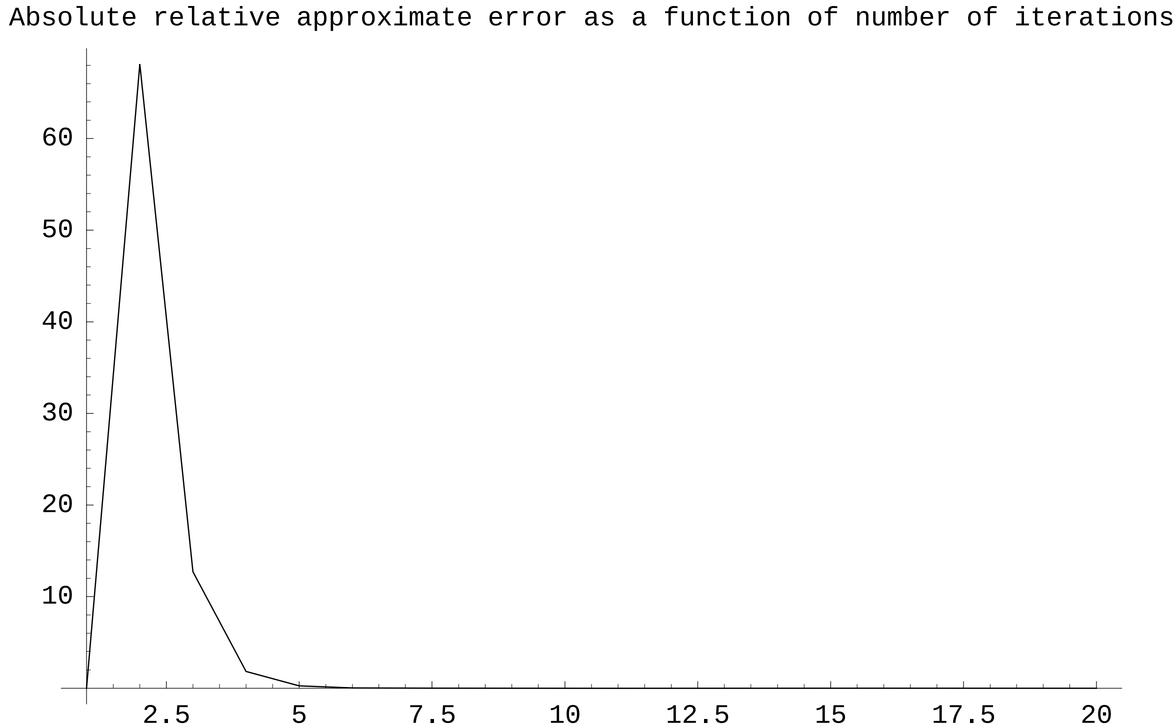
In[38]:=
```
Eaplot = Table[Ea[i], {i, 2, n_maximum, 2}];
```

In[39]:=
```
ListPlot[Eaplot, PlotJoined → True,
 PlotRange -> All, AxesOrigin -> {1, Min[Eaplot]}, PlotLabel ->
  "Absolute approximate error as a function of number of iterations"];
```



In[40]:=
```
ϵaplot = Table[ϵa[i], {i, 2, nmaximum, 2}];
```

In[41]:=
```
ListPlot[ϵaplot, PlotJoined → True,
 PlotRange -> All, AxesOrigin -> {1, Min[ϵaplot]},
 PlotLabel -> "Absolute relative approximate error
   as a function of number of iterations"];
```



In[42]:=
```
sigdigplot = Table[sigdig[i], {i, 2, n_maximum, 2}];
```

In[43]:=
```
<< Graphics`Graphics`
```

In[44]:=
```
BarChart[sigdigplot];
```