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

Trapezoidal rule is based on the Newton-Cotes formula that if one approximates the integrand of the integral by an nth order polynomial, then the integral of the function is approximated by the integral of that nth order polynomial. Integration of polynomials is simple and is based on the calculus. Trapezoidal rule is the area under the curve for a first order polynomial (straight line) that approximates the integral. [click here for textbook notes] [click here for power point presentation].

The following simulation illustrates the Trapezoidal 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 limits of the integration *a* and *b* respectively. By entering this data, the program will calculate the exact value of the integral, followed by the results using Trapezoidal rule for *n* = 1, 2, 3, and 4 segments. The program will also 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[124]:= f[x_] := 300.0 * x / (1.0 + Exp[x]);
```

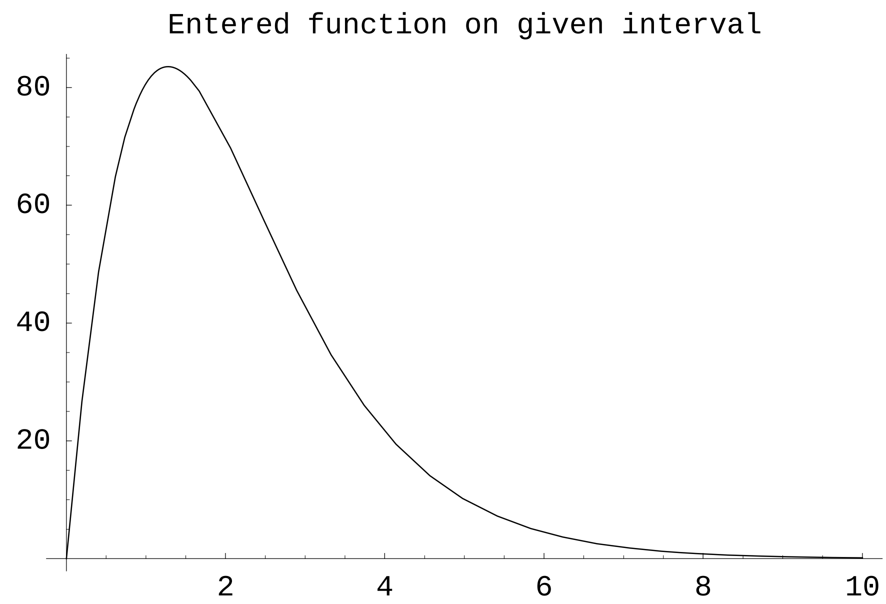
Lower and upper limit of the integral, *a* and *b* respectively.

```
In[125]:= a = 0.0;
          b = 10.0;
```

Maximum number of iterations. This number must be even.

```
In[127]:= n_maximum = 40;
```

In[128]:=
```
curve = Plot[f[x], {x, a, b}, PlotLabel → "Entered function on given interval",
   TextStyle → {FontSize → 11}];
General::spelll Off;
```



■ True Solution

This is the solution found by Mathematica

In[130]:=
```
Actual = Integrate[f[x], {x, a, b}]
```

Out[130]=
```
246.59
```

■ Value of integral as a function of iterations

Trapezoidal Rule

In[131]:=
```
Array[AV, n_maximum];
```

In[132]:=
```
For[i = 1, i ≤ n_maximum, i++, If[i ≤ 1, AV[i] = (f[a] + f[b]) / 2 * (b - a),
   h = (b - a) / i; AV[i] = h / 2 * (f[a] + 2 * Sum[f[a + j * h], {j, 1, i - 1}] + f[b])]]
```

■ Absolute true error

In[133]:=
```
Array[Et, n_maximum];
```

In[134]:=
```
For[i = 1, i <= n_maximum, i++, Et[i] = Abs[Actual - AV[i]]]
```

■ Absolute relative true error

In[135]:=
```
Array[ϵt, n_maximum];
```

In[136]:=
```
For[i = 1, i <= n_maximum, i++, ϵt[i] = Abs[Et[i] / Actual * 100]]
```

■ Absolute approximate error

In[137]:=
```
Array[Ea, n_maximum];
```

In[138]:=
```
For[i = 1, i <= n_maximum, i++, If[i <= 1,  Ea[i] = 0, Ea[i] = Abs[AV[i] - AV[i - 1]]]]
```

■ Absolute relative approximate error

In[139]:=
```
Array[ϵa, n_maximum];
```

In[140]:=
```
For[i = 1, i <= n_maximum, i++, If[i <= 1, ϵa[i] = 0, ϵa[i] = Abs[Ea[i] / AV[i] * 100]]]
```

■ Significant digits at least correct

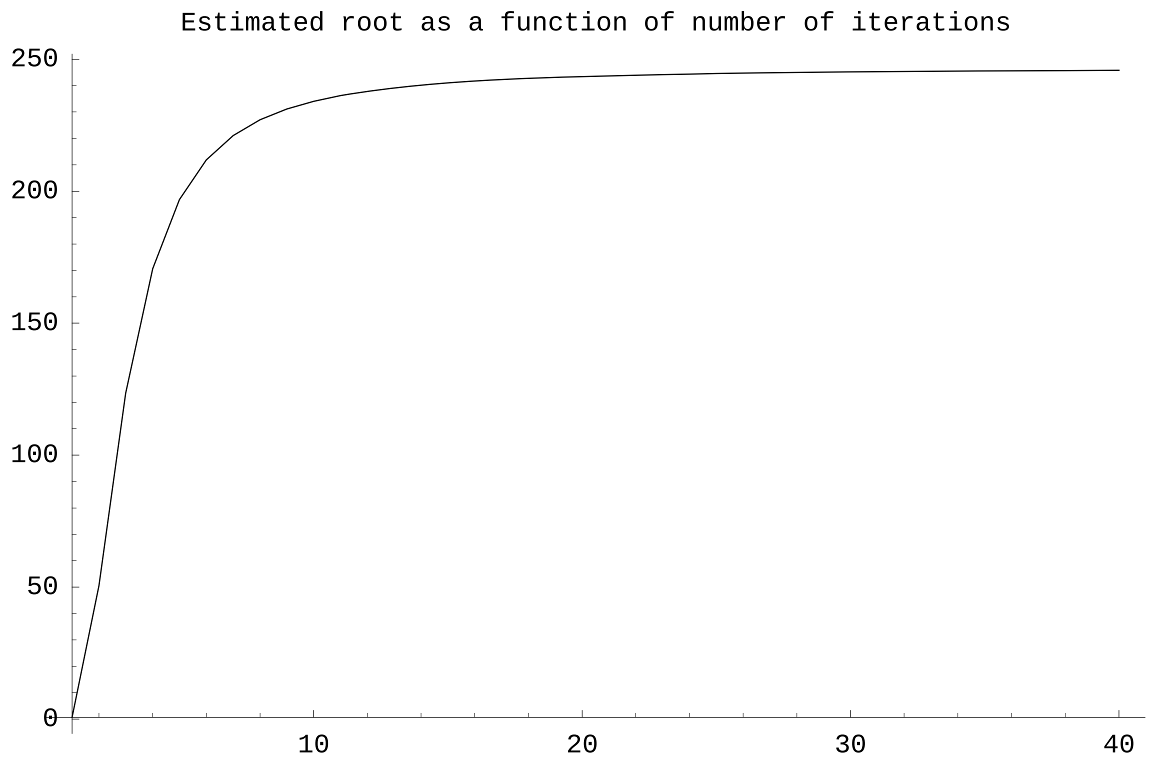
In[141]:=
```
Array[sigdig, n_maximum];
```

In[142]:=
```
For[i = 1, i <= n_maximum, i++, If[(ϵa[i] ≥ 5) || (i <= 1),
  sigdig[i] = 0, sigdig[i] = Floor[(2 - Log[10, Abs[ϵa[i] / 0.5]])]]]
```

■ Graphs

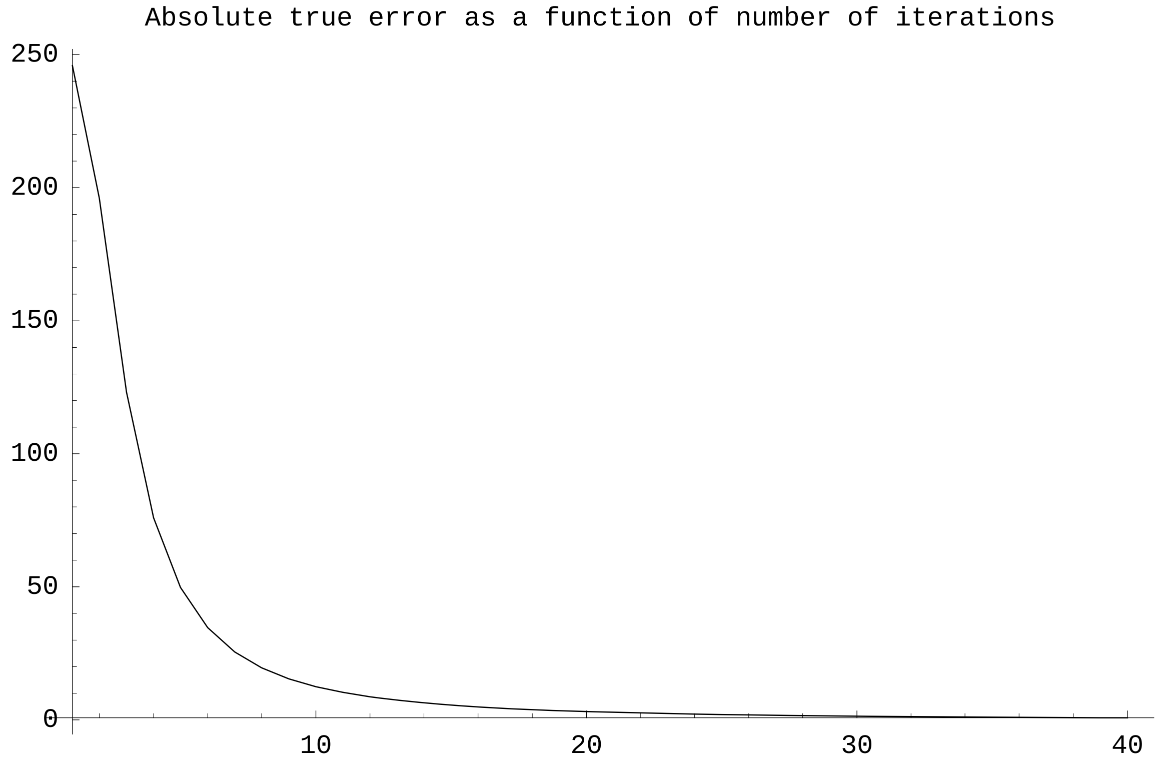
In[143]:=
```
xrplot = Table[AV[i], {i, 1, n_maximum}];
```

In[144]:=
```
ListPlot[xrplot, PlotJoined → True,
 PlotRange -> All, AxesOrigin -> {1, Min[xrplot]},
 PlotLabel -> "Estimated root as a function of number of iterations"];
```



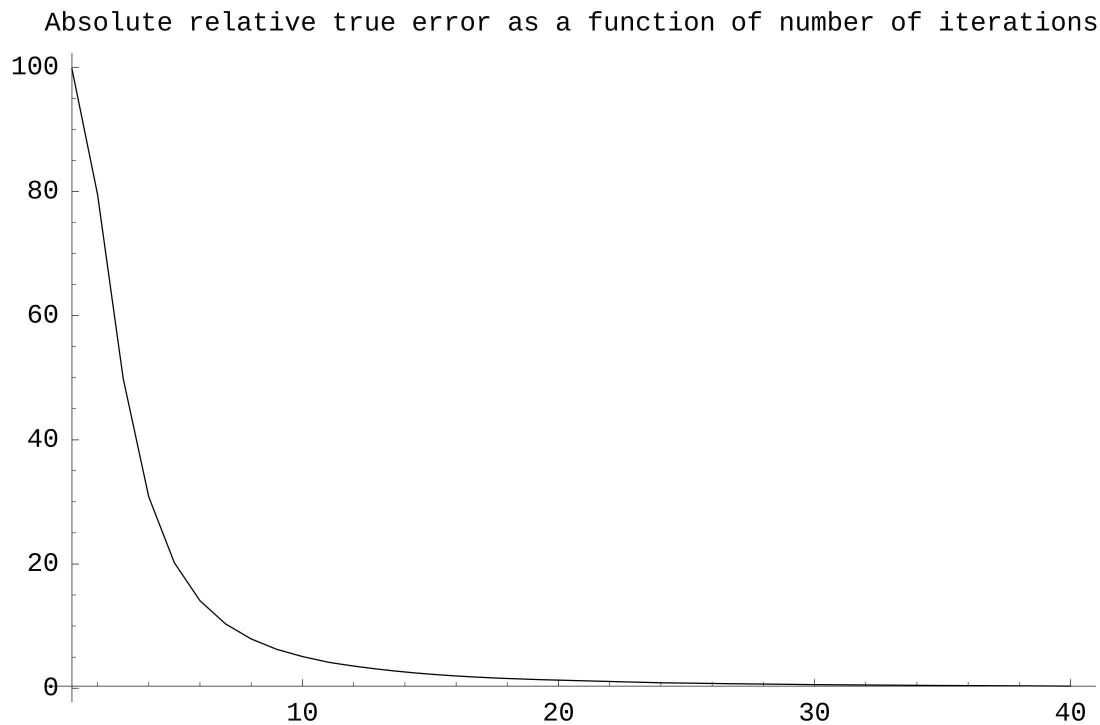
In[145]:=
```
Etplot = Table[Et[i], {i, 1, n_maximum}];
```

```
In[146]:= ListPlot[Etplot, PlotJoined → True,
           PlotRange -> All, AxesOrigin -> {1, Min[Etplot]},
           PlotLabel -> "Absolute true error as a function of number of iterations"];
```



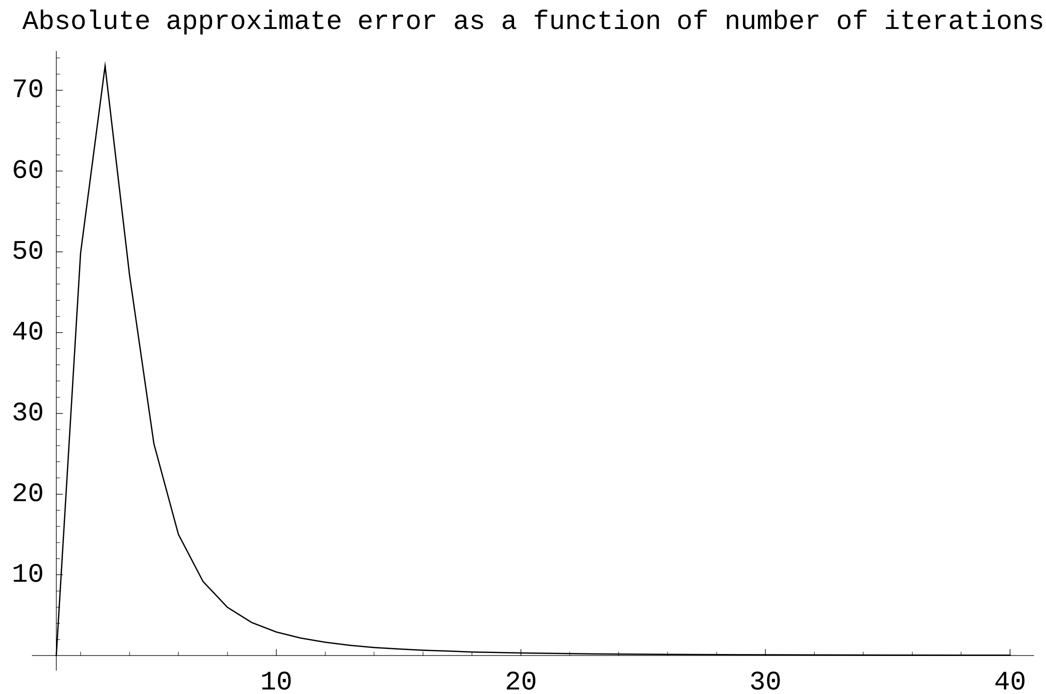

```
In[147]:= ∈tplot = Table[∈t[i], {i, 1, nmaximum}];
```

```
In[148]:= ListPlot[∈tplot, PlotJoined → True,
  PlotRange -> All, AxesOrigin -> {1, Min[∈tplot]}, PlotLabel ->
   "Absolute relative true error as a function of number of iterations"];
```



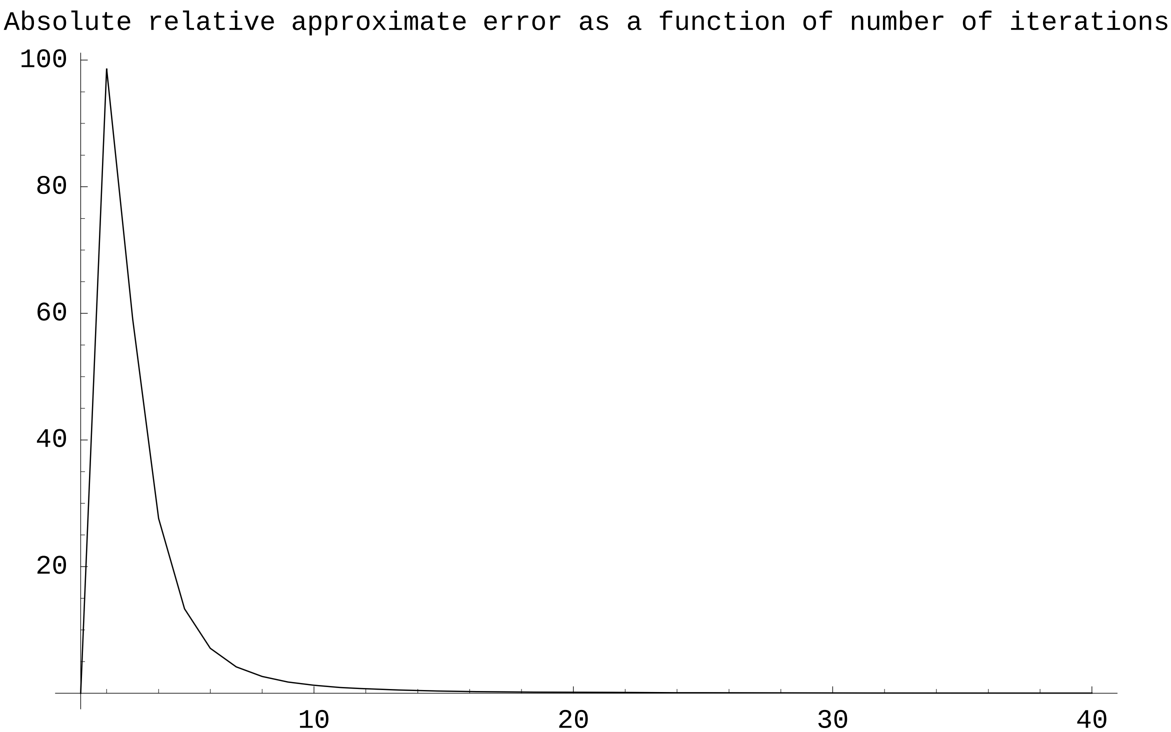

```
In[149]:= Eaplot = Table[Ea[i], {i, 1, nmaximum}];
```

In[150]:=
```
ListPlot[Eaplot, PlotJoined → True,
 PlotRange -> All, AxesOrigin -> {1, Min[Eaplot]}, PlotLabel ->
  "Absolute approximate error as a function of number of iterations"];
```



In[151]:=
```
ϵaplot = Table[ϵa[i], {i, 1, n_maximum}];
```

In[152]:=
```
ListPlot[∈aplot, PlotJoined → True,
 PlotRange -> All, AxesOrigin -> {1, Min[∈aplot]},
 PlotLabel -> "Absolute relative approximate error
   as a function of number of iterations"];
```



In[153]:=
```
sigdigplot = Table[sigdig[i], {i, 1, nmaximum}];
```

In[154]:=
```
<< Graphics`Graphics`
```

In[155]:= **BarChart[sigdigplot];**